



UN/CEFACT

United Nations Centre for Trade Facilitation and Electronic Business

Techniques and Methodologies
Group (TMG)

UN/CEFACT's Modeling Methodology (UMM)

UMM Meta Model – Foundation Module

Candidate for 2.0

Public Draft

NOT FOR IMPLEMENTATION

2008-09-10

1	Table of Contents	
2	Table of Contents	2
3	1 About this Document	4
4	1.1 Status of this Document	4
5	1.2 Revision History	4
6	1.3 Document Context	4
7	2 Project Team.....	5
8	2.1 Disclaimer	5
9	2.2 Contact	5
10	2.3 Project Team Participants	6
11	3 Introduction.....	6
12	3.1 Audience.....	6
13	3.2 Related Documents	7
14	3.3 UN/CEFACT's Modeling Methodology (UMM): Overview.....	7
15	3.4 Objectives	9
16	3.4.1 Goals of the Technical Specification	9
17	3.4.2 Requirements	9
18	3.4.3 Caveats and Assumptions.....	9
19	3.5 Structure of the UMM Foundation Module	10
20	4 Dependencies on other UMM Modules (normative).....	11
21	4.1 Abbreviations of Stereotypes	11
22	4.2 Dependency between Base Module and Foundation Module.....	11
23	5 UMM Foundation Module.....	12
24	5.0 Foundation Module Management	12
25	5.0.1 Abbreviations of Stereotypes	12
26	5.0.2 Conceptual Description (informative)	13
27	5.0.3 Stereotypes and Tag Definitions (normative).....	14
28	5.0.4 Constraints (normative).....	16
29	5.1 Business Requirements View.....	16
30	5.1.0 Sub-Views in the Business Requirements View.....	16
31	5.1.1 Business Domain View.....	19
32	5.1.2 Business Partner View	34
33	5.1.3 Business Entity View	37
34	5.2 Business Choreography View	44

35	5.2.1	Sub-Views in the Business Choreography View	44
36	5.2.2	Business Transaction View	48
37	5.2.3	Business Collaboration View	68
38	5.2.4	Business Realization View	86
39	5.3	Business Information View	91
40	5.3.1	Abbreviations of Stereotypes	91
41	5.3.2	Conceptual Description (informative)	91
42	5.3.3	Stereotypes and Tag Definitions (normative).....	92
43	5.3.4	Constraints (normative).....	92
44	5.3.5	Example using UPCC (UML Profile for Core Components) and CCMA (Core Components	
45		Message Assembly) (informative)	93
46		Copyright Statement	95
47			
48			

49 1 About this Document

50 1.1 Status of this Document

51

52 This document is currently being developed, and is at Open Development Process Step 5: “Public Review”

53 (i) Activities

54 The UNECE secretariat provides links on the UNECE website to the Public Draft and related
55 information. The FMG notifies Heads of Delegation and various e-mail distribution list
56 subscribers that the Public Draft is available for review and provides them with review-process
57 details. The project team processes comments and posts updated Public Drafts and comment
58 logs to the PG website or the UNECE website (through the Secretariat). The
59 comment/update/posting cycle continues until the PG approves a project team
60 recommendation to conclude ODP5. While the criteria, evaluation, and ultimate decision to
61 conclude ODP5 is left to the PG, the PG must ensure that the project team has met all comment
62 processing requirements (see Annex I). The draft resulting from this step is called an
63 Implementation Draft.

64 (ii) Artifacts

65 Typical artifacts produced by ODP5 include:

- 66 a. Implementation Draft
- 67 b. Comment log.

68 The first review cycle of ODP 5 is open until 1st September 2008. Please send your comments to the Project
69 Lead Christian Huemer (huemer@big.tuwien.ac.at)

70 1.2 Revision History

Version	Release	Date	Comment
Candidate for 2.0	Internal Draft	2008-04-11	
Candidate for 2.0	Public Draft	2008-06-27	Changes are marked in green

71 1.3 Document Context

72 The UMM meta model is divided into a set of meta modules. This means that the UMM meta model is
73 partitioned into functional levels, ranging from core, minimal functionality, to complete functionality. The
74 following partition levels have been defined for meta modules:

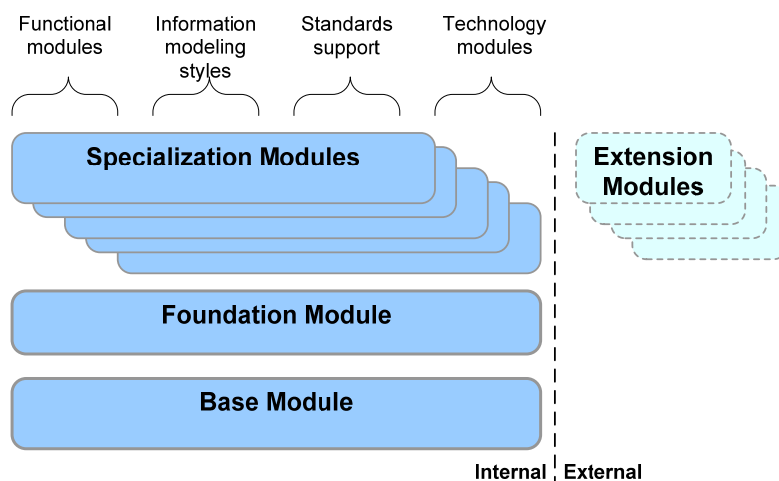


Figure 1 Module structure of the UMM meta model

Base: Covers the fundamental principles that are shared across all the other modules.

Foundation: Includes the core concepts of the UMM. In addition, it defines all the concepts that are used as part of the minimal methodology to produce a UMM compliant business collaboration model. Furthermore, it provides fundamental principles which are shared across all other modules.

Specialization: Multiple specialization modules might define add-on concepts to the foundation. Each specialization module addresses a specialized type of analysis that extends the foundation module at a well-defined extension point for a specific topic. Specialization modules might become candidates for later inclusion into the foundation module.

Extension: Extension modules serve the same purpose as specialization modules. Whereas specialization modules are developed and maintained by UN/CEFACT, extension modules are adding features that are created and maintained by organization(s) which are external to UN/CEFACT.

This specification defines the foundation module of UMM 2.0.

2 Project Team

2.1 Disclaimer

The views and specification expressed in this document are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this technical specification.

2.2 Contact

Name:	Christian Huemer
Company:	Vienna University of Technology
Street:	Favoritenstrasse 9-11/188
City, state, zip/other:	1040 Vienna
Nation:	Austria
Phone:	+43 1 58801 18882
Email:	huemer@big.tuwien.ac.at

2.3 Project Team Participants

Project Team Lead:	Christian Huemer	Austria
Editing Team:	Jens Dietrich	Germany
	Birgit Hofreiter	Austria
	Christian Huemer	Austria
	Philipp Liegl	Austria
	Glenn Miller	Canda
	Harry Moyer	Australia
	Rainer Schuster	Austria
	Marco Zapletal	Austria
Contributors:	Steve Capell	Australia
	Sylvie Colas	France
	Barbara Flügge	Switzerland
	William McCarthy	USA
	Thomas Motal	Austria
	Christian Senf	Germany
	Nita Sharma	USA
	Gunther Stuhec	Germany

The Editing Team of this UMM foundation module likes to thank former members of TMG's Business Process Working Group (BPWG) who have spent enormous efforts in putting the UMM into a stage that we were able to build upon in order to create this foundation module.

3 Introduction

3.1 Audience

A reader of the document MUST have a deep understanding of UML 2.1.2. She or he MUST be able to understand meta models denoted as UML class diagrams. She or he SHOULD be familiar with the UML 2.1.2. meta model, at least she or he MUST be able to check back the UML 2.1.2. meta model. The reader SHOULD be familiar with OCL 2.0 in order to understand the OCL constraints of this UMM profile – those who are not familiar with OCL are provided with a plain text description of the constraint.

The information described in this manual is aimed at

- advanced business process modelers who check a UML model for UMM compliance (if not supported by a tool)
- advanced business process modelers who train other business process modelers and business process analysts
- software designers who want to produce UML tools providing support for this UMM foundation module

- software designers who want to produce tools to transform UMM compliant business collaboration models into specifications within an IT-layer (ebXML, Web Services, UN/EDIFACT, etc.).
- software designers who want to produce repositories to register UMM compliant business collaboration models

3.2 Related Documents

- **UN/CEFACT**
 - o UN/CEFACT Open Development Process (TRADE/R.650/Rev.4/Add.1/Rev.1 - 19 April 2007)
http://www.unece.org/cefact/cf_plenary/plenary07/trd_R650_Rev4_A1E.pdf
 - o UPCC: UML Profile for Core Components
<http://unstandards.org:8080/display/public/UPCC++UML+Profile+for+Core+Components>
 - o Core Component Technical Specification
http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf
- **International Organization for Standardization (ISO)**
 - o Open-edi Reference Model. ISO/IEC 14662
[http://standards.iso.org/ittf/PubliclyAvailableStandards/c037354_ISO_IEC_14662_2004\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c037354_ISO_IEC_14662_2004(E).zip)
- **Object Management Group (OMG)**
 - o Unified Modeling Language Specification (UML), Version 2.1.2
<http://www.omg.org/docs/formal/07-02-05.pdf>

3.3 UN/CEFACT's Modeling Methodology (UMM): Overview

UN/CEFACT's Modeling Methodology (UMM) is a UML modeling approach to design the business services that each partner must provide in order to collaborate. It provides the business justification for the services to be implemented in a service-oriented collaboration architecture. Thus, a primary vision of UN/CEFACT is to capture the business knowledge that enables the development of low cost software based on service-oriented architectures (SOA) helping the small and medium size companies (SMEs), and emerging economies to engage in e-Business practices. UMM focuses on developing a global choreography of inter-organizational business processes and their information exchanges. UMM models are notated in UML syntax and are platform independent models. The platform independent UMM models identify which services need to be realized in a service-oriented architecture, implementing the business collaboration. This approach provides insurance against technical obsolescence.

The UMM, as described in this document, is the formal description technique for describing any Open-edi scenario as defined in ISO/IEC 14662 "Open-edi reference model". An Open-edi scenario is a formal means to specify a class of business transactions having the same business goal, such as, purchasing or inventory management. The primary scope of UMM is the Business Operations View (BOV) and not the Functional Service View (FSV) as defined in ISO/IEC IS 14662. The BOV is defined as "a perspective of business transactions limited to those aspects regarding the making of business decisions and commitments among organizations", while the FSV is focused on implementation specific, technological aspects of Open-edi. The commitments of the BOV layer are reflected in the choreography of the inter-organizational business processes and their information exchanges. At the FSV layer, this choreography must be implemented by a set of composite services. Therefore it follows, that UMM, which targets the BOV layer, defines what the business is about; and the technologies on the FSV layer define how to implement the business by a service-oriented architecture.

This version of the UMM consists of three views each covering a set of well defined artifacts:

- Business Requirements View (bRequirementsV)

- Business Choreography View (bChoreographyV)
- Business Interaction View (bInteractionV)

Business Requirements View (bRequirementsV): The Business Requirements View is used to gather existing knowledge. It identifies the business processes in the domain and the business problems that are important to stakeholders. It is important at this stage that business processes are not constructed, but discovered. Stakeholders might describe intra-organizational as well as inter-organizational business processes. All of this takes place in the language of the business experts and stakeholders. The business requirements view results in a categorization of the business domain (manifested as a hierarchical structure of packages) and a set of relevant business processes (manifested as use cases). The result may be depicted in use case diagrams. In order to model the dynamics of each business process, one may use a Business Process Activity Model, or a Sequence Diagram, which would be placed beneath the Business Process Use Case. As a practical note, the Business Process Activity Model may depict a process or processes which involve one or more Business Partners. A Sequence Diagram will depict information exchanges between two or more Business Partners. The Business Partners are described within their own package (Business Partner View). A Business Process Activity Model may show state changes to Business Entities. Business Entities are “real-world” things having business significance and are shared among the business partners involved in the collaboration. The Business Entities and their lifecycles of state changes are modeled in the Business Entity View. Furthermore, the Business Entity View also contains one or more packages which represent the conceptual data structures of the Business Entities.

Business Choreography View (bChoreographyV): The Business Choreography View is used to define and document the global choreography between collaborating business partners in an inter-organizational business process. Within the Business Choreography View, the Business Transaction View contains and documents the requirements of Business Transaction Use Cases, and their participating Authorized Roles. The dynamics of a Business Transaction Use Case are described by a Business Transaction. A business transaction defines a simple choreography of exchanging business information between two authorized roles and an optional response. A business transaction identifies the business actions of each partner responsible for sending and receiving the business information. These actions correspond to the requirements of any solution that must be implemented on each business partner’s side in a service-oriented collaboration architecture. Within the Business Choreography View, the Business Collaboration View contains and documents the requirements of Business Collaboration Use Cases and their participating Authorized Roles. The dynamics of a Business Collaboration Use Case are described by a Business Collaboration Protocol. A Business Collaboration Protocol choreographs the flow among business transactions, and/or nested Business Collaboration Protocols. This flow depends on the states of business entities. When a Business Collaboration Use Case is identified, but different sets of parties may execute this collaboration, the different Realizations (executions) may be modeled within the Business Realization View, as a Business Realization Use Cases.

Business Information View (bInformationV): An execution of a business transaction usually results in the change of state of one or more business entities. Thus, the information exchanged in a transaction should be limited to the minimum information needed to change the state of a business entity. Nevertheless, UMM allows the definition of an information exchange in a document-centric approach – even if this is not recommended. A Business Information View contains Business Information Artifacts. UMM does not mandate a specific Business Information Modeling approach. However, UMM strongly recommends that

228 Business Information is modeled in accordance to UN/CEFACT's Core Components Technical Specification
229 and Message Assembly Guidelines. In order to model Core Components by means of UML, UN/CEFACT
230 provides the Profile for Core Components (UPCC).

231 **3.4 Objectives**

232 **3.4.1 Goals of the Technical Specification**

233 The goals of this specification are:

- 234 • To define the semantics of well-formed UMM business collaboration models, which describe a public
235 choreography of an inter-organizational system. Local choreographies and private processes of a
236 businesses partner are out of scope.
- 237 • To define the validation rules for UMM compliant business collaboration models.
- 238 • To clarify the basic concepts that a UMM-compliant business collaboration model is based on.
- 239 • To provide an unambiguous definition for UMM business collaboration models that allows an
240 unambiguous mapping to artifacts for deployment in a service-oriented architecture. Note, that the
241 mapping itself is not part of UMM.
- 242 • To define a UML profile for the UMM foundation module that allows UML tool vendors to customize
243 their tools to be UMM compliant. Better UML 2.1.1. tool support will lead to a growing UMM user
244 base.

245 **3.4.2 Requirements**

246 This specification is guided by the following key requirements derived from the above goals:

- 247 • The UMM foundation module defines only those modeling concepts that are considered as
248 fundamental to deliver a UMM compliant model. Additional advanced modeling concepts shall be
249 covered in specialization and extension modules.
- 250 • The UMM foundation module is directed towards the Business Operational View of Open-edi. This
251 means it is independent of certain implementation technologies used in SOAs like Web Services and
252 ebXML or other future technologies. However, the UMM compliant business collaboration models
253 must be defined in a way that allows a mapping to an implementation technology of choice. Such a
254 mapping is not part of the UMM foundation module. It is a candidate for a specialization/extension
255 module.
- 256 • Today, the UML is the most commonly supported modeling language by modeling tools. In order to
257 use the broad range of tools, a UMM business collaboration model must be a special kind of UML
258 model. Thus, the UMM foundation module is based on the UML meta model. In fact, it provides a
259 UML Profile consisting of stereotypes, tagged definitions and constraints.
- 260 • In order to support a broad adoption of the UMM-modeling approach, the formal descriptions of the
261 UMM is supplemented by a set of examples that show UMM compliant artifacts.

262 **3.4.3 Caveats and Assumptions**

263 This specification makes the following assumptions:

- 264 • This UML profile is based on the UML meta-model version 2.1.2. This version is the current OMG
265 version. Using another UML meta-model as a basis for the development of a UMM compliant
266 business collaboration model may not deliver correct results.
- 267 • The basic concepts of the UMM and the way they relate to each other are described and explained
268 by means of a meta model (to be found in the non-normative "conceptual description" sections of
269 this document).

- Different specialization and extension modules might extend the foundation module in order to define additional semantics to the minimum semantics required to create a UMM compliant business collaboration models.

3.5 Structure of the UMM Foundation Module

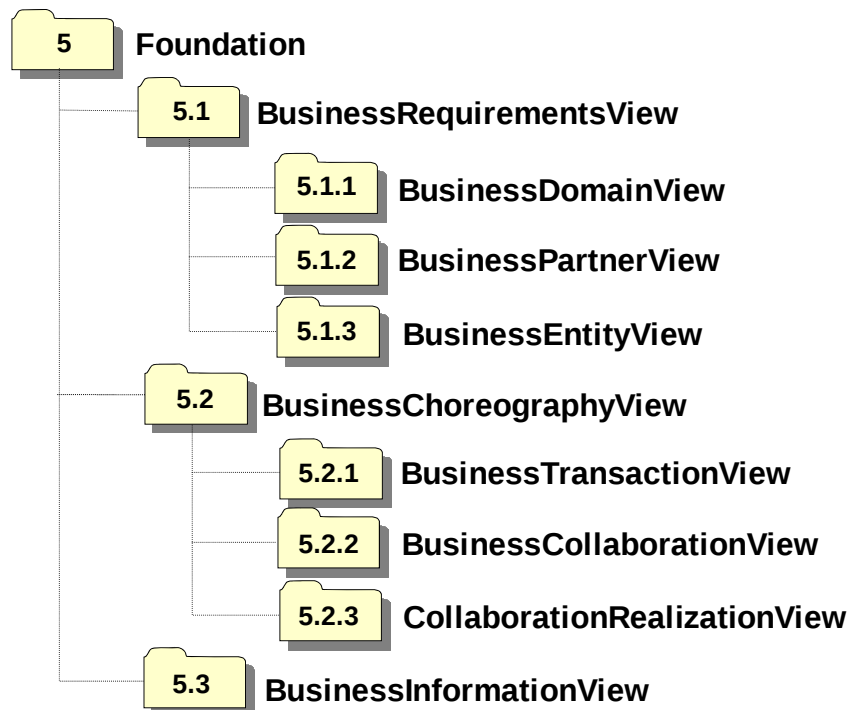


Figure 2 Package overview of UMM Foundation Module meta model

Section 5 defines the UML profile of the foundation module of the UMM meta model (Section 1, Figure 1). The figure above (Figure 2), shows the package structure of the foundation module of the UMM meta model. The numbers referring to the subsections are included in figure 2. Those numbers refer to the subsections containing the stereotypes, tag definitions and constraints of the corresponding package. The first level packages of the foundation module conform to the three views of the current UMM version: Business Requirements View (5.1), Business Choreography View (5.2), and Business Information View (5.3).

The Business Requirements View (5.1) comprises the Business Domain View (5.1.1), the Business Partner View (5.1.2), and the Business Entity View (5.1.3). The second top-level package, the Business Choreography View (5.2) consists of the Business Transaction View (5.2.1), the Business Collaboration View (5.2.2), and the Collaboration Realization View (5.2.3). The third top-level package is the Business Information View (5.3). It does not contain any sub packages.

Each section describing a package is structured in the same way. The first subsection is informative. It describes the conceptual model of the artefact that is addressed by the package. The second subsection is normative and defines all the stereotypes and associated tag definitions that are defined in the package. The third subsection is normative and includes all the constraints, both in plain text and in OCL, that apply to the respective package. The two remaining informative subsections cover on the one hand side worksheets used to gather information from business people in order to create the UMM models and on the other hand side examples to depict instances of the artefact type addressed by the package.

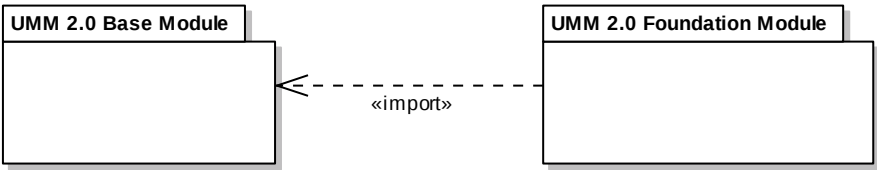
294 **4 Dependencies on other UMM Modules (normative)**

295 **4.1 Abbreviations of Stereotypes**

Stereotype Abbreviation	Full Stereotype Name
bInformation	BusinessInformation
bLibrary	BusinessLibrary
InfEnvelope	InformationEnvelope

296

297 **4.2 Dependency between Base Module and Foundation Module**



298

299 **Figure 3 UMM Foundation Dependencies**

300 The UMM foundation module 2.0 is built on top of the UMM base module 2.0. This means that all
301 stereotypes and tag definitions defined in the UMM base module 2.0 are imported into the UMM
302 foundation module 2.0. The figure below shows the stereotypes defined in the UMM base module also used
303 in the foundation module. Note that the stereotypes of the base module are identified with notes in all
304 figures of this specification. The formal definition of the stereotypes *bInformation*, *InfEnvelope* and *bLibrary*
305 is given in the UMM base module 2.0 specification. In the foundation module, packages - that are containers
306 of stereotypes realizing main UMM artefacts - are defined as specializations of the base stereotype *bLibrary*.
307 This means that such packages and their contents are candidates for registration in a registry. In the UMM
308 foundation module 2.0 we do not define any stereotype that directly inherits from *bLibrary*. As a
309 consequence, only packages are candidates for registration.

310 The concepts of *bInformation* and *InfEnvelope* are used to define the business document information being
311 exchanged between authorized roles in a UMM business transaction.

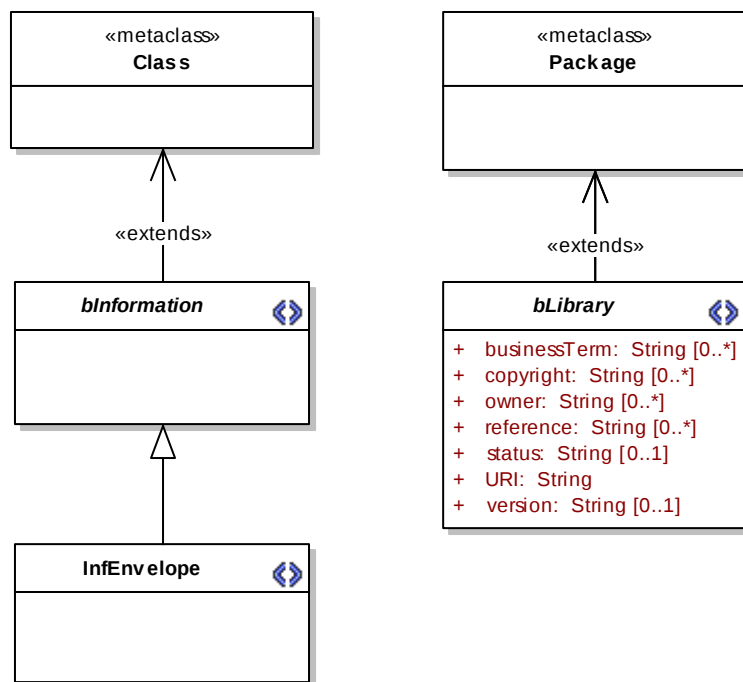


Figure 4 UMM Base Abstract Syntax

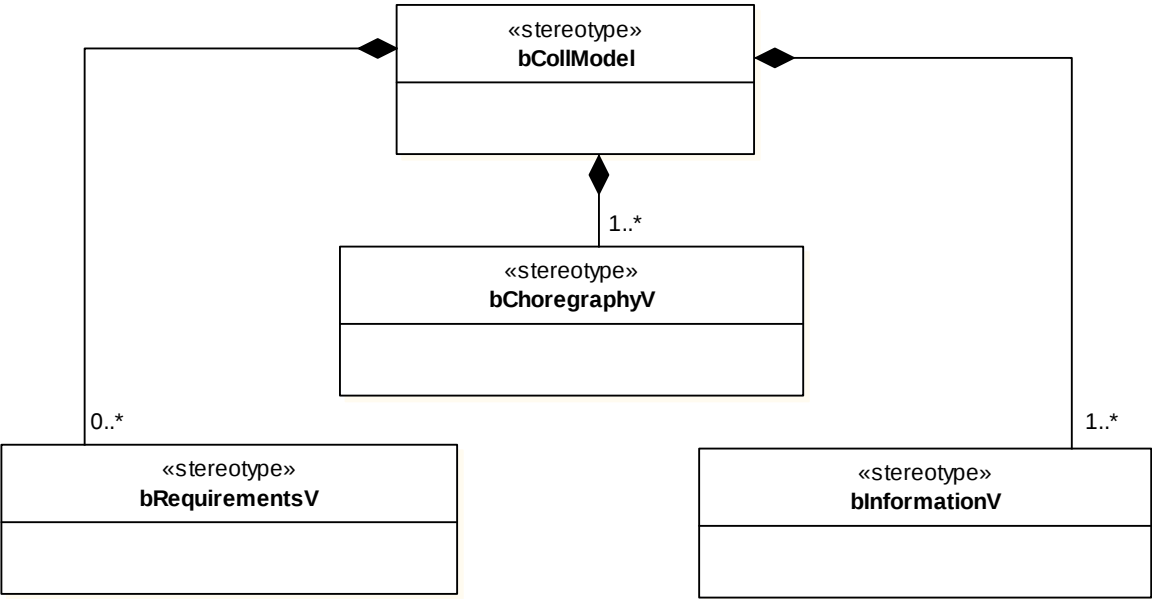
5 UMM Foundation Module

5.0 Foundation Module Management

5.0.1 Abbreviations of Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bCollModel	BusinessCollaborationModel
bRequirementsV	BusinessRequirementsView
bChoreographyV	BusinessChoreographyView
bInformationV	BusinessInformationView
bLibrary	BusinessLibrary

319 5.0.2 Conceptual Description (informative)



320

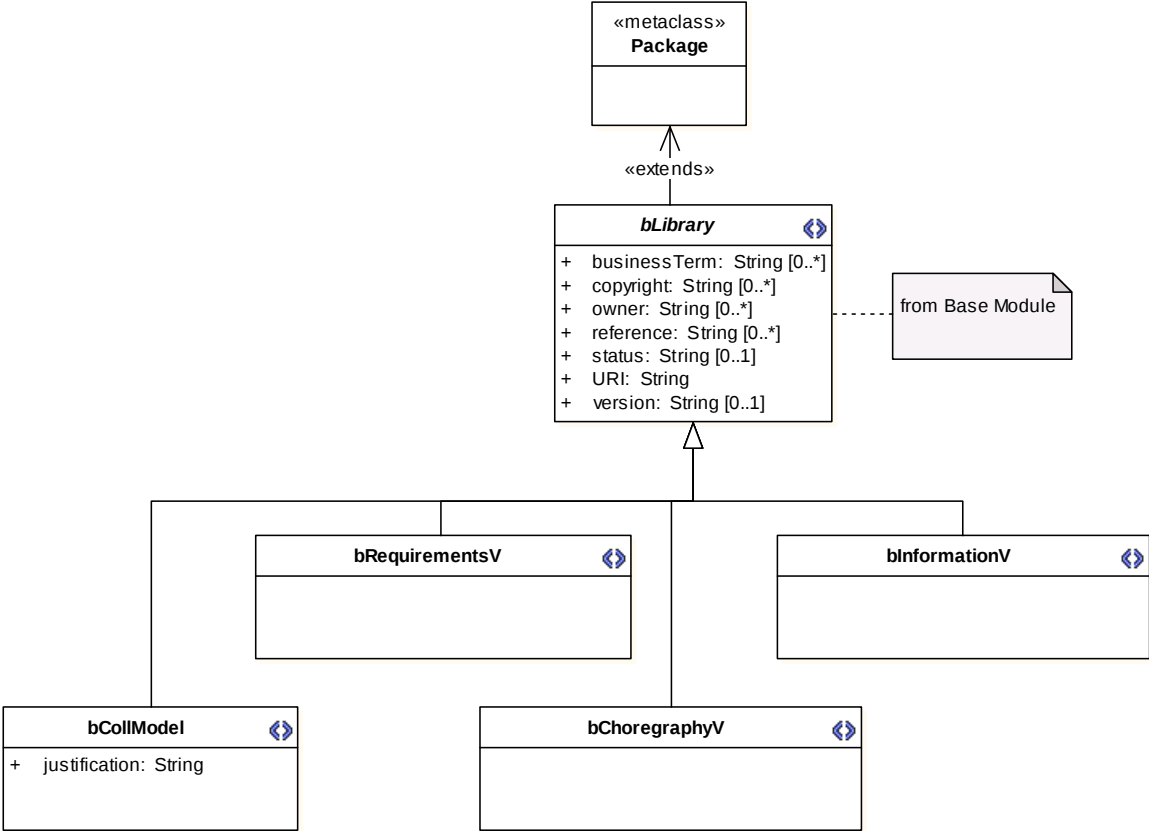
321 **Figure 5 UMM Foundation Module Management - Conceptual Overview**

322 A project that follows the UMM approach leads to a business collaboration model. A business collaboration
323 model that is UMM compliant is stereotyped as *bCollModel*. As described above, the UMM is built by three
324 views. The business requirements view stereotyped as *bRequirementsV* is optional and may occur multiple
325 times in a business collaboration model. The business choreography view (stereotyped as *bChoreographyV*)
326 and the business information view (stereotyped as *bInformationV*) are mandatory parts of a business
327 collaboration model and may also occur multiple times.

328 Within the business requirements view the specific requirements of the business collaboration between two
329 or more business partners are captured. The collected information from the business collaboration is then
330 further elaborated within the business choreography view. The information exchanged during the process is
331 modeled in the business information view. For further information on the specific sub-views of the UMM,
332 please see the relevant sub-chapters of this specification.

333

334 5.0.3 Stereotypes and Tag Definitions (normative)



335

336 Figure 6 UMM Foundation Module Management - Abstract Syntax

Stereotype	BusinessCollaborationModel	
Base Class	Package	
Parent	BusinessLibrary (from Base Module)	
Description	A business collaboration model is a model that is compliant to the UMM meta model. It MUST be compliant to the base and foundation module, and it MAY be compliant to one or more specialization and/or extension modules. Since a business collaboration model is of base class package, a UML model MAY contain one to many business collaboration models. Therefore, either the root element of a UML model is stereotyped as business collaboration model or any of the packages beneath the root element.	
Tag Definition	justification	
	Type	String
	Multiplicity	1
	Description	Explains the reason from a business perspective why the given business case is considered for possible business collaborations.
	Inherited tagged values: – businessTerm – copyright	

337

	– owner – reference – status – URI – version
--	--

338

Stereotype	BusinessRequirementsView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business requirements view is a container for all elements needed to identify and describe the requirements of collaborations between business partners. It captures the relevant packages which are used for discovering relevant business processes and their business partners executing/participating in them, as well as the lifecycle and state changes of business entities which are important within a business process.
Tag Definition	Inherited tagged values: – businessTerm – copyright – owner – reference – status – URI – version

339

Stereotype	BusinessChoreographyView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business choreography view is a container for all elements needed to describe the choreography of business collaborations at various levels.
Tag Definition	Inherited tagged values: – businessTerm – copyright – owner – reference – status – URI – version

Stereotype	BusinessInformationView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business information view is a container for all elements representing the exchanged information in business collaborations.

Tag Definition	Inherited tagged values:
	– businessTerm – copyright – owner – reference – status – URI – version

340

341 5.0.4 Constraints (normative)

342

343 Constraints with respect to the *BusinessCollaborationModel* (bCollModel)

344 C.1. A *BusinessCollaborationModel* MUST contain one to many *BusinessChoreographyViews*

345 C.2. A *BusinessCollaborationModel* MUST contain one to many *BusinessInformationViews*

346 C.3. A *BusinessCollaborationModel* MAY contain zero to many *BusinessRequirementsViews*

347 C.4. A *BusinessRequirementsView*, a *BusinessChoreographyView* and a *BusinessInformationView* MUST
348 be directly located under a *BusinessCollaborationModel*

349 5.1 Business Requirements View

350 5.1.0 Sub-Views in the Business Requirements View

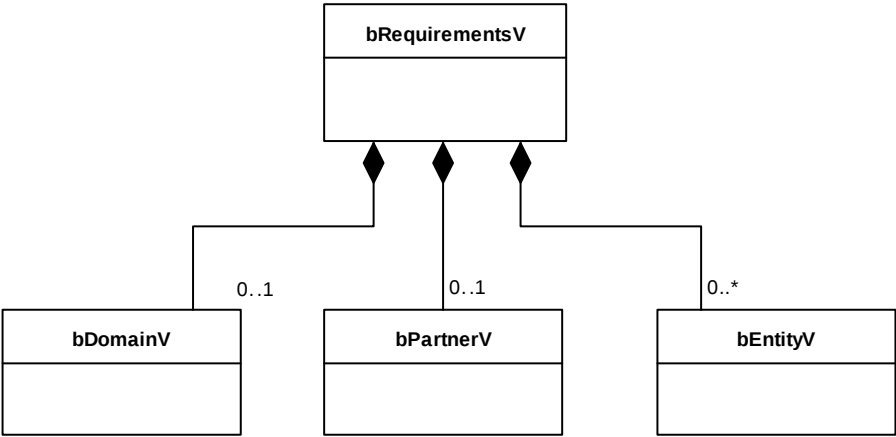
351 5.1.0.1 Abbreviations and Stereotypes

352

Stereotype Abbreviation	Full Stereotype Name
bRequirementsV	BusinessRequirementsView
bDomainV	BusinessDomainView
bPartnerV	BusinessPartnerView
bEntityV	BusinessEntityView

353

354 5.1.0.2 Conceptual Description (informative)



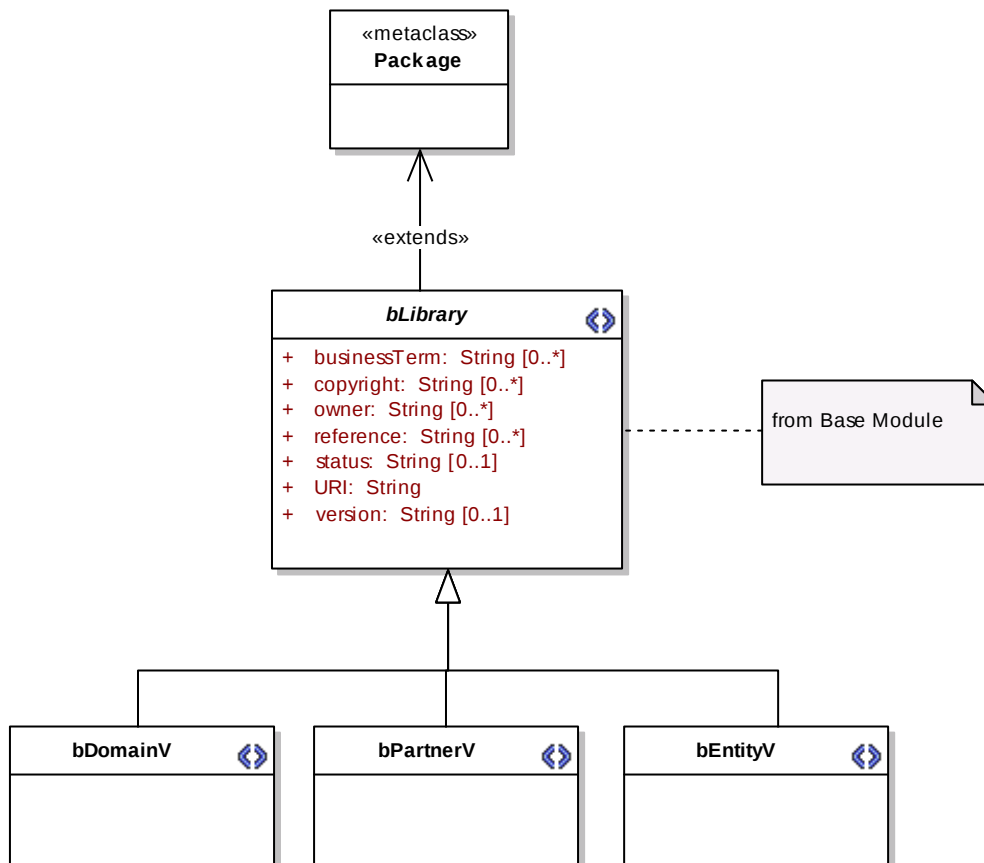
355

356

Figure 7 BusinessRequirementsView Conceptual Overview

357 The *BusinessRequirementsView* is composed by three significant sub-views which are all of optional use.
358 Firstly, the *BusinessDomainView* captures all of the business processes which may be of interest for the
359 domain under consideration. In order to enable users to readily identify business processes, these business
360 processes are classified into business categories. This classification is done by creating business areas and
361 process areas. However, UN/CEFACT recommends using this classification, but it is not mandatory. Secondly,
362 the *BusinessPartnerView* specifies a list of business partners and stakeholders that are involved in the
363 business processes defined in the *BusinessDomainView*. Furthermore the relationships between each others
364 are defined in this package. Finally, the *BusinessEntityView* defines the business entities that are involved in
365 a business process. A business entity is a real-world thing having business significance that is shared among
366 two or more business partner in a collaborative business process (e.g. order, account, etc.). It is important to
367 depict the possible state changes of such business entities within the *BusinessEntityView* in order to get an
368 understanding of how a collaborative business process affects such real-world things during the execution of
369 a business process.

370 **5.1.0.3 Stereotypes and Tag Definitions (normative)**



371
372

Figure 8 BusinessRequirementsView Abstract Syntax

Stereotype	BusinessDomainView
Base Class	Package
Parent	BusinessLibrary (from Base Module)

Description	The business domain view is used to discover business processes that are of relevance in a project. A business domain is a framework for identification and understanding of business processes as well as for categorizing them according to a classification schema. The business domain view is a container capturing the categorization scheme and categorized business processes.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

373

Stereotype	BusinessPartnerView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business partner view captures a list of business partners and stakeholders in the domain under consideration as well as the relationships between them.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

374

Stereotype	BusinessEntityView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business entity view is a container to describe the lifecycle of a business entity having business significance in the modelled domain including its' business entity states.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

375

376

- 377
- 378
- 379
- 380
- 381
- 382
- 383
- 384
- 385
- 386
- 5.1.0.4 Constraints (normative)
- Constraints with respect to a BusinessRequirementsView:
- C.5. A BusinessRequirementsView MAY contain zero or one BusinessDomainView.
- C.6. A BusinessRequirementsView MAY contain zero or one BusinessPartnerView.
- C.7. A BusinessRequirementsView MAY contain zero to many BusinessEntityViews.
- C.8. A BusinessDomainView, a BusinessPartnerView, and a BusinessEntityView MUST be located directly under a BusinessRequirementsView.

385

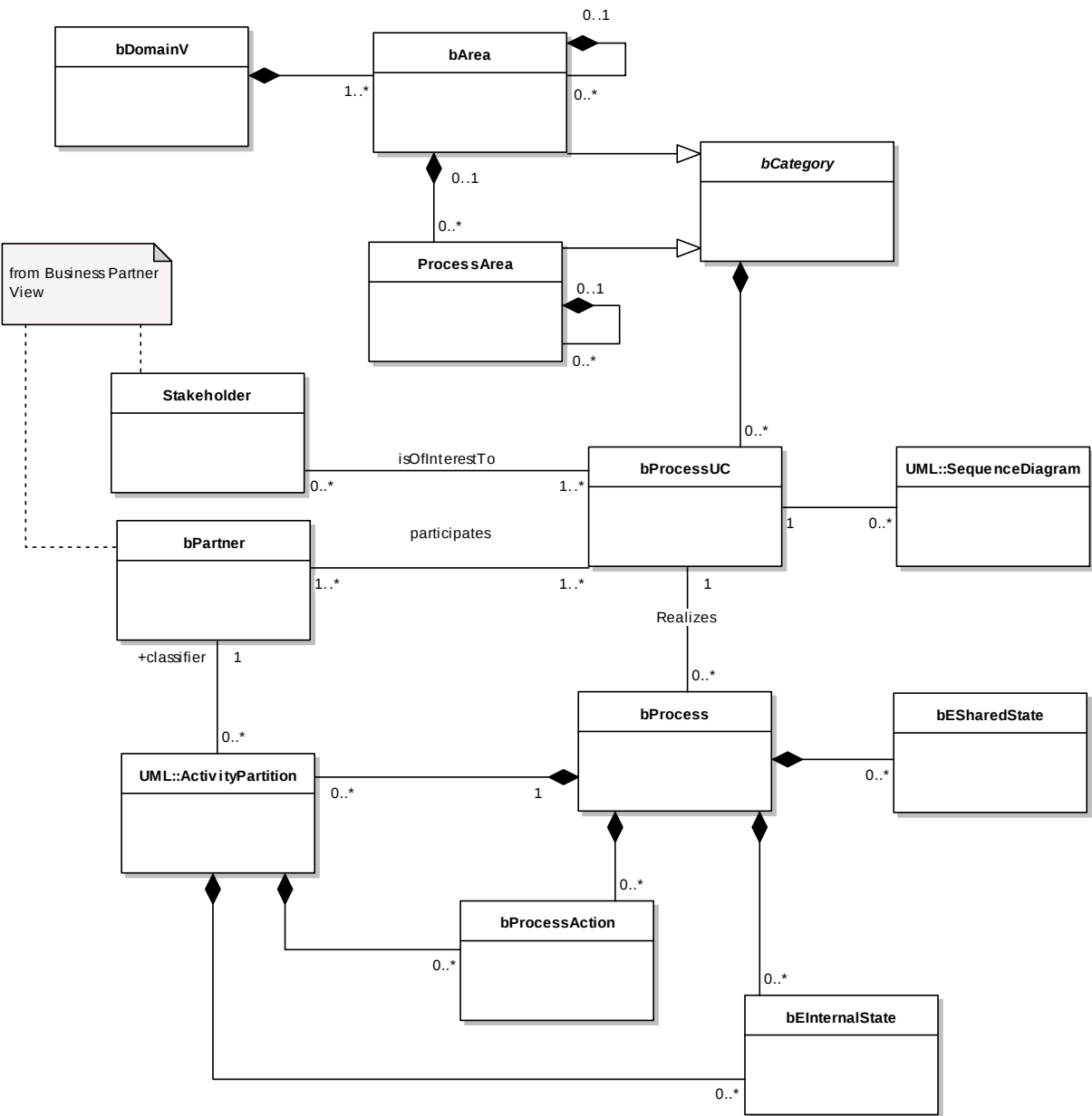
5.1.1 Business Domain View

386

5.1.1.1 Abbreviations of Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bDomainV	BusinessDomainView
bCategory	BusinessCategory
bArea	BusinessArea
ProcessArea	ProcessArea
bProcessUC	BusinessProcessUseCase
bProcess	BusinessProcess
bProcessAction	BusinessProcessAction
bESharedState	SharedBusinessEntityState
bEInternalState	InternalBusinessEntityState

387



389

390 **Figure 9 BusinessDomainView – Conceptual Overview**

391 The business domain view is used to discover business processes use cases that are of relevance in a project.
392 A business process use case is executed by at least one (but possibly more) business partners. A business
393 partner might execute multiple business process use cases. Thus, the *participates* association between
394 *BusinessPartner* and *BusinessProcessUseCase* is a (1..n) to (0..n) association. A stakeholder does not need to
395 participate in a business process use case. A stakeholder might have interest in multiple business process use
396 cases and a business process use case might be of interest to multiple stakeholders. The relationship
397 between a *BusinessProcessUseCase* and a *Stakeholder* is described by the *isOfInterestTo* dependency in
398 UMM. A business process can be decomposed into sub-processes using the «include» and «extends»
399 association stereotypes.

400 To enable users to readily identify business process use cases, they should be classified into business
401 categories. A business category is an abstract concept, which has two concrete specializations – business
402 area and process area. A business area corresponds to a division of an organization and a process area
403 corresponds to a set of common operations within the business area. A business area might be composed of
404 other business areas. This means, a business area may form a hierarchy. Thus, a unary (0..1) to (0..n)
405 composition is defined for a *BusinessArea*. The lowest level of a business area hierarchy includes process
406 areas or business processes use cases. Therefore, we have a (0..1) to (0..n) composition between
407 *BusinessArea* and *ProcessArea*. Furthermore, a *BusinessArea* may also include 0..n *BusinessProcessUseCases*
408 if no further classification using process areas is required. Similar to a business area, a process area may form
409 a hierarchy. This means, a unary (0..1) to (0..n) composition is defined for a *ProcessArea*. Similar to a
410 business area, a *ProcessArea* may contain zero to many *BusinessProcessUseCases*. On the lowest level of a
411 *ProcessArea* hierarchy, at least one *BusinessProcessUseCase* must be present.

412 The flow of a business process use case may be described by business processes. Thus, a
413 *BusinessProcessUseCase* is realized by zero to many *BusinessProcesses*. A business process represents the
414 dynamic behavior of a business process use case. A business process corresponds to a flow of actions
415 performed by one participant or even by more participants. If two or more business partners collaborate, a
416 business process is divided into partitions – one for each business partner. In case of an internal business
417 process, which is executed by one partner only, a single partition for that partner is optional. Consequently,
418 a *BusinessProcess* is composed of zero or more UML *ActivityPartitions*. An *ActivityPartition* is assigned to one
419 *BusinessPartner*; a *BusinessPartner* is assigned to one *ActivityPartition*. However, a *BusinessPartner* may be
420 assigned to multiple *ActivityPartitions* – each one in a different *BusinessProcess*. Hence, there is a 1 to (0..n)
421 association between *BusinessPartner* and *ActivityPartition*.

422 A business process is described as a flow of business process actions. In the case where no activity partition
423 is used, the business process actions are directly included in the Activity Diagram of the business process. In
424 case of activity partitions, a business process action is assigned to the partition of the business partner
425 executing the action. The need for a collaborative business process is identified whenever a transition
426 connecting two business process actions crosses activity partitions. It follows, that either a *BusinessProcess* is
427 composed of one or more *BusinessProcessAction* or an *ActivityPartition* (which is part of a business process)
428 is composed of one or more *BusinessProcessActions*. A business process action might be refined by another
429 business process. Thus a *BusinessProcessAction* is composed of zero or one *BusinessProcess* which in turn is a
430 composite of zero or one *BusinessProcessActions*.

431 A business process may also denote important states of business entities that are manipulated during the
432 execution of a business process. A business entity state is the output from one business action and input to
433 another business action. There is a transition from a business process action to a business entity state
434 signaling an output as well as a transition from a business entity state to a business process action signaling
435 an input. Some business entity states are meaningful to one business partner only. These are internal
436 business entity states. Business entity states that must be communicated to a business partner are shared
437 business entity states. A business process may include both internal and shared business entity states.
438 Hence, a *BusinessProcess* is composed of zero to many *InternalBusinessEntityStates* and of zero to many
439 *SharedBusinessEntityStates*. If a business process uses activity partitions, the two business process actions
440 creating and consuming an internal business entity state are in the same activity partition. In contrast, the
441 two business process actions creating and consuming a shared business entity state are in different activity
442 partitions. A shared business entity state signals the need for a collaborative business process.

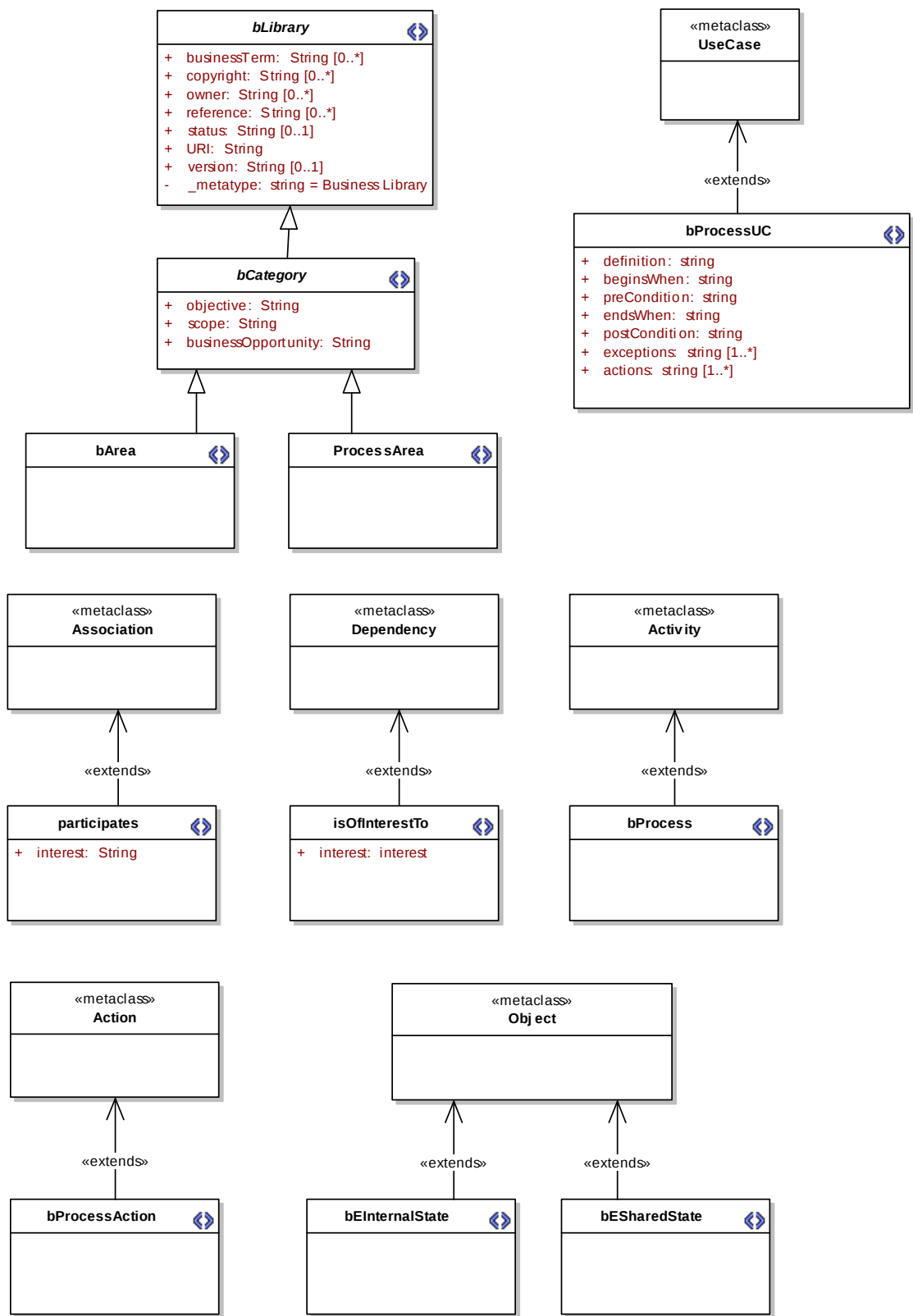


Figure 10 BusinessDomainView Abstract Syntax

Stereotype	<i>BusinessCategory (abstract)</i>	
Base Class	Package	
Parent	BusinessLibraryPackage (from Base Module)	
Description	A business category is an abstract concept. Business categories are used to classify the business processes in the Business Domain View. The prime purpose of classifying the business processes is to enable potential users to readily identify processes that have been defined in the business category under consideration. Consequently a business category is used to group either other business categories or business processes that belong to the respective business category. The Business Domain View is structured by its specializations <i>BusinessArea</i> and <i>ProcessArea</i> (see below for these stereotype definitions).	
Tag Definition	objective	
	Type	String
	Multiplicity	1
	Description	The purpose to be achieved by the business process within the business category under consideration.
	scope	
	Type	String
	Multiplicity	1
	Description	Defines the boundaries of the business category under consideration.
	businessOpportunity	
	Type	String
	Multiplicity	1
	Description	The strategic interest from a business perspective in order to address the business category under consideration.
	Inherited tagged values: – baseURN – owner – copyright – reference – version – status – businessTerm	

447

Stereotype	<i>BusinessArea</i>
Base Class	Package
Parent	BusinessCategory
Description	A business area usually corresponds to a division of an enterprise. Business areas might be structured recursively. A business area is a category of decomposable business areas or process areas (on the lowest

	level of business area hierarchy). This means that a business area collates either other business areas, process areas or business process use case. The UMM does not mandate a specific classification schema. A classification schema that might be used is the Porter Value Chain. Based on the Porter Value Chain, the UN/CEFACT Common Business Process Catalog recommends a list of eight flat (i.e. non-recursive) categories: Procurement/Sales, Design, Manufacture, Logistics, Recruitment/Training, Financial Services, Regulation, and Health Care. This list of business areas is considered as non exhaustive.
Tag Definition	Inherited tagged values: – objective – scope – businessOpportunity – baseURN – owner – copyright – reference – version – status – businessTerm

448

Stereotype	ProcessArea
Base Class	Package
Parent	BusinessCategory
Description	A process area corresponds to a set of common operations within a business area. Process areas might be structured recursively. A process area is a category of common business process use cases. This means a process area collates either other process areas or business process use cases. The UMM does not mandate a specific classification schema. The UN/CEFACT Common Business Process Catalog recommends a list of five flat (i.e. non-recursive) categories that correspond to the five successive phases of business collaborations as defined by the ISO Open-edi model: Planning, Identification, Negotiation, Actualization, Post-Actualization.
Tag Definition	Inherited tagged values: – objective – scope – businessOpportunity – baseURN – owner – copyright – reference – version – status – businessTerm

449

Stereotype	BusinessProcessUseCase
Base Class	UseCase
Parent	N/A

Description	A business process use case is a set of related activities that together create value for a business partner. A business process use case might be performed by a single business partner type or by multiple business partner types crossing organizational boundaries. In the case where organizations collaborate in a business process, the business process should create value for all of its participants. A business process use case can be decomposed into sub-processes using the «include» and «extends» association stereotypes defined in UML.	
Tag Definition	definition	
	Type	String
	Multiplicity	1
	Description	Gives a definition of the business process use case. This definition must describe the customer value to be created by the business process use case. In the case of a business process use case executed by multiple parties, it describes the value to be created to all participants.
	beginsWhen	
	Type	String
	Multiplicity	1
	Description	Specifies a business event that triggers the initiation of the business process use case.
	preCondition	
	Type	String
	Multiplicity	1
	Description	Specifies a condition that has to be fulfilled in order to execute a business process use case. This condition SHOULD refer to states in the life cycle of a business entity. A pre-condition statement MAY use Boolean operators specifying a combination of multiple business entity states.
	endsWhen	
	Type	String
	Multiplicity	1
	Description	Specifies a business event that leads to the termination of the business process use case.
	postCondition	
	Type	String
	Multiplicity	1
	Description	Specifies a condition that will be reached after executing the business process use case. Usually, this condition SHOULD refer to states in the life cycle of a business entity. A post-condition statement MAY use Boolean operators specifying a combination of multiple business entity states.
	exceptions	

	Type	String
	Multiplicity	1..*
	Description	Identifies situations leading to a deviation of the regular execution of the business process use case.
	actions	
	Type	String
	Multiplicity	1..*
	Description	Lists the tasks that together make up a business process use case. In the case of a business process use case executed by multiple parties, a special emphasis on interface tasks is needed. An interface task is a step in the business process use case that requires communication with another business partner.

450

Stereotype		Participates
Base Class	Association	
Parent	N/A	
Description	Describes the association between a business partner and a business process use case. This stereotype defines that the business partner provides input to and/or output from the associated business process use case.	
Tag Definition	interest	
	Type	String
	Multiplicity	1
	Description	Describes the vested interest of the business partner type in the business process associated by this participates-association.

451

Stereotype		isOfInterestTo
Base Class	Dependency	
Parent	N/A	
Description	Describes a dependency from a business process use case to a stakeholder. This stereotype defines that a business process use case depends on the interest of the connected stakeholder.	
Tag Definition	interest	
	Type	String
	Multiplicity	1
	Description	Describes the vested interest of the stakeholder in the business process use case linked by this is of interest to dependency.

452

453

Stereotype	BusinessProcess
Base Class	Activity
Parent	N/A
Description	The business process describes the behavior of a business process use case between the involved business partners. It is a tool to identify requirements to collaborate between two or more business partners. A business process refines a business process use case by describing its dynamic behaviour.
Tag Definition	No tagged values.

454

Stereotype	BusinessProcessAction
Base Class	Action
Parent	N/A
Description	A business process action corresponds to a step in the execution of a business process. A business process action might be refined by another business process. In this case, a UML call behavior action MUST be used as base class for the business process action
Tag Definition	No tagged values.

455

Stereotype	InternalBusinessEntityState
Base Class	ObjectNode
Parent	N/A
Description	The internal business entity state represents a state of a business entity that is internal to the business process of a business partner.
Tag Definition	No tagged values.

456

Stereotype	SharedBusinessEntityState
Base Class	ObjectNode
Parent	N/A
Description	The shared business entity state represents a state of a business entity that is shared between the business processes between two involved business partners.
Tag Definition	No tagged values.

Form for Business Domain View	
General	
Name	
Description	
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Business Area(s)	
Business Area No 1	
Business Area No 2	
Business Area No 3	
Business Area No 4	
Business Area No 5	
Business Area No 6	(add columns as needed)

Form for Business Area	
General	
Name	
Description	
Details	

Objective	
Scope	
Business Opportunity	
Included in	(insert the parent Business Area or Business Domain View)
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Business Area(s)	(insert additional nested business areas if appropriate; otherwise fill process areas that apply)
Business Area No 1	
Business Area No 2	
Business Area No 3	
Business Area No 4	
Business Area No 5	(add columns as needed)
Process Area(s)	(you only fill process areas if you do not have completed a business area above)
Process Area No 1	
Process Area No 2	
Process Area No 3	
Process Area No 4	
Process Area No 5	(add columns as needed)

Form for Process Area	
General	
Name	
Description	
Details	
Objective	
Scope	
Business Opportunity	
Included in	(insert the parent Business Area or Process Area)
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Process Area(s)	(if present)
Process Area No 1	
Process Area No 2	
Process Area No 3	
Process Area No 4	
Process Area No 5	(add columns as needed)

460

461

Form for Business Process	
General	
Name	
Description	
Details	
Classified to Business Areas and Process Areas	
Participants and their interests	
Stakeholders and their interests	
Reference(s)	
Start / End Characteristics	
Pre-condition	
Post-condition	
Begins When	
Ends When	
Actions	
Exceptions	
Relationships	
Included Business Processes	
Affected Business Entities	

463

464 **5.1.1.5 Constraints (normative)**465 **C.9. A BusinessDomainView MUST include one to many BusinessAreas.**466 **C.10. A BusinessArea MUST include one to many BusinessAreas or one to many ProcessAreas or**
467 **one to many BusinessProcessUseCases.**468 **C.11. A ProcessArea MUST contain one to many other ProcessAreas or one to many**
469 **BusinessProcessUseCases**

- C.12. A *BusinessProcessUseCase* MUST be associated with one to many *BusinessPartners* using the *participates* relationship
- C.13. A *BusinessProcessUseCase* MAY be associated with zero to many *Stakeholders* using the *isOfInterestTo* relationship
- C.14. A *BusinessProcessUseCase* SHOULD be refined by zero to many *BusinessProcesses*
- C.15. A *BusinessProcess* MUST be modeled as a child of a *BusinessProcessUseCase*
- C.16. A *BusinessProcessUseCase* MAY be refined by zero to many UML Sequence Diagrams
- C.17. A *BusinessProcess* MAY contain zero to many *ActivityPartitions*
- C.18. A *BusinessProcess*, which has no *ActivityPartitions*, MUST contain one or more *BusinessProcessActions* and MAY contain zero to many *InternalBusinessEntityStates* and zero to many *SharedBusinessEntityStates*.
- C.19. An *ActivityPartition* being part of a *BusinessProcess* MUST contain one to many *BusinessProcessActions* and MAY contain zero to many *InternalBusinessEntityStates*.
- C.20. A *SharedBusinessEntityStates* MUST NOT be located in an *ActivityPartition*. (They must be contained within the *BusinessProcess* even if this *BusinessProcess* contains *ActivityPartitions*.)

5.1.1.6 Example (informative)

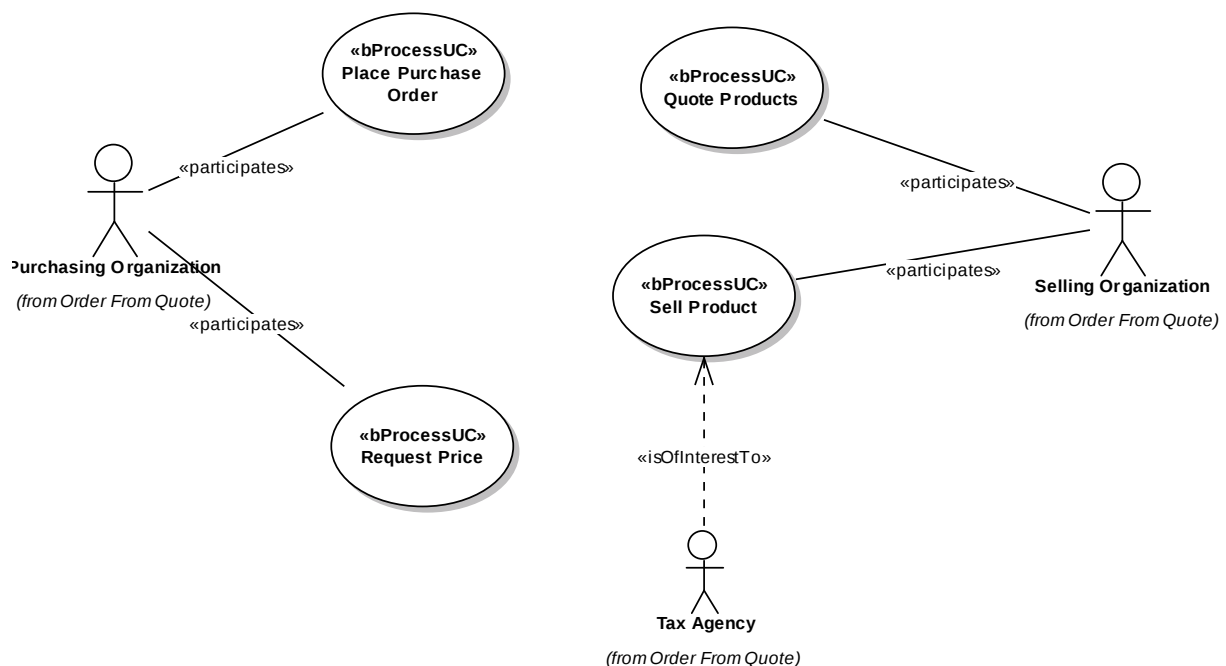


Figure 11 Business Domain View Example: Negotiation In the Order from Quote Example (Use Case Diagram showing Business Process Use Cases)

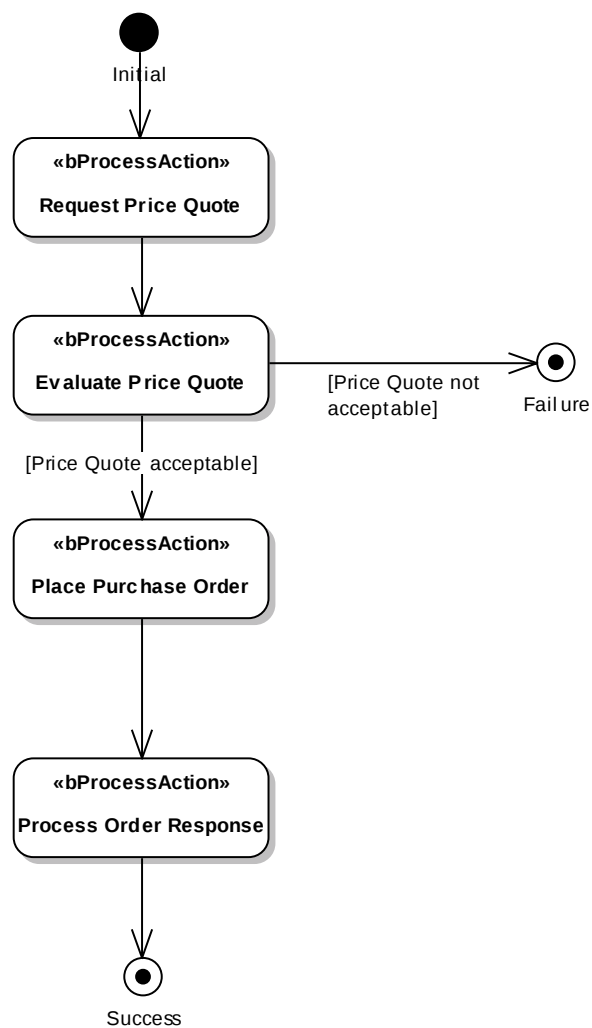


Figure 12 Business Domain View Example: Business Process of the internal Place Order Business Process Use Case (Activity Diagram)

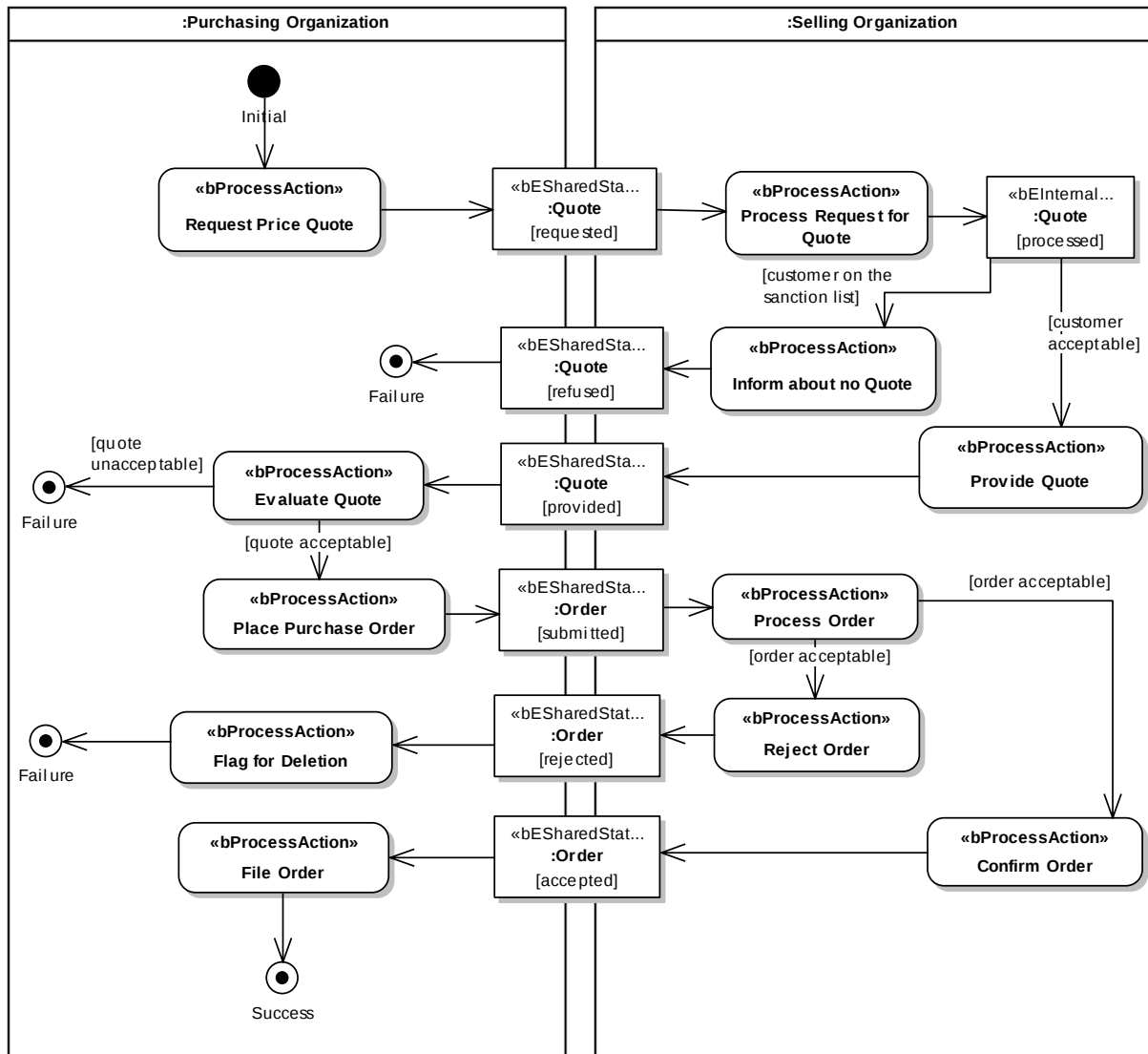
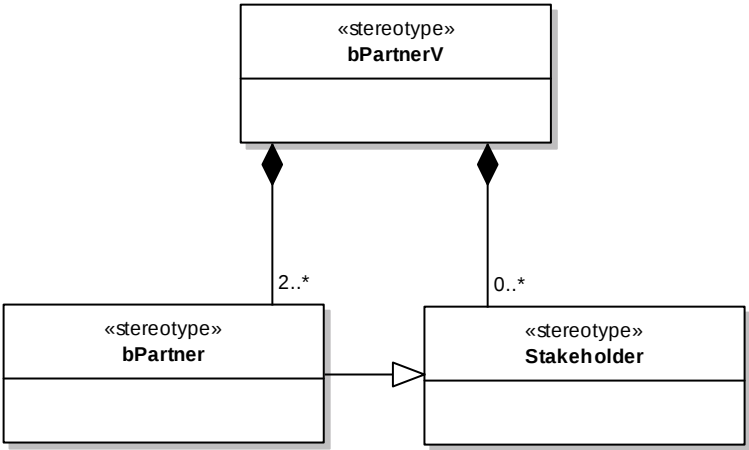


Figure 13 Business Domain View Example: Business Process of the to-be-designed inter-organizational process called Purchase Product (Activity Diagram)

5.1.2 Business Partner View

5.1.2.1 Abbreviations of Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bPartnerV	BusinessPartnerView
bPartner	BusinessPartner
Stakeholder	Stakeholder

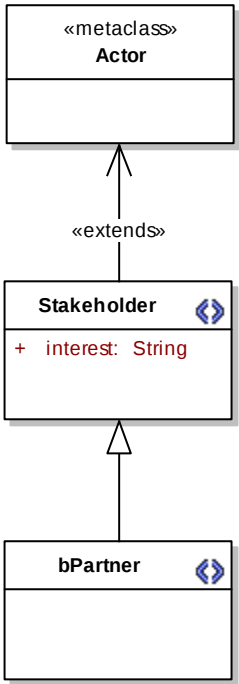


499

500 **Figure 14 BusinessPartnerView – Conceptual Overview**

501 A business partner is an organization type, an organizational unit type or a person type that participates in a
502 business process. A *BusinessPartnerView* must contain at least two *BusinessPartners*. A stakeholder is a
503 person or representative of an organization who has a stake – a vested interest – in a certain business
504 category or in the outcome of a business process. By definition, a business partner always has a vested
505 interest in the business processes which they are participating in. Therefore, a *BusinessPartner* is a special
506 type of a *Stakeholder*. In UML, specific relationships between Actors MAY be defined. The business partner
507 view does not restrict the definition of those relationships between business partners and/or stakeholders.
508 For example, generalizations between business partners MAY be defined.

509 5.1.2.3 Stereotypes and Tag Definitions (normative)



510

511 **Figure 15 BusinessPartnerView – Abstract Syntax**

Stereotype	Stakeholder	
Base Class	Actor	
Parent	N/A	
Description	A stakeholder is a person or representative of an organization who has a stake – a vested interest – in a certain business category or in the outcome of a business process. A stakeholder does not necessarily participate in the execution of a business process.	
Tag Definition	interest	
	Type	String
	Multiplicity	1
	Description	Describes the vested interest of the stakeholder in the business category it is defined within.

512

Stereotype	BusinessPartner	
Base Class	Actor	
Parent	Stakeholder	
Description	A business partner type is an organization type, an organizational unit type or a person type that participates in a business process. Business partner types typically provide input to and/or receive output from a business process. Due to the fact that a business partner type participates in a business process, they have, by default, a vested interest in the business process. Therefore, a business partner type is a special kind of stakeholder.	
Tag Definition	Inherited tagged values: - interest	

513 5.1.2.4 Constraints (normative)

- 514 C.21. A *BusinessPartnerView* MUST contain at least two to many *BusinessPartners*. If the
- 515 *BusinessPartnerView* is hierarchically decomposed into subpackages these *BusinessPartners* MAY be
- 516 contained in any of these subpackages.
- 517 C.22. A *BusinessPartnerView* MAY contain zero to many Stakeholders

518 5.1.2.5 Example (informative)



519

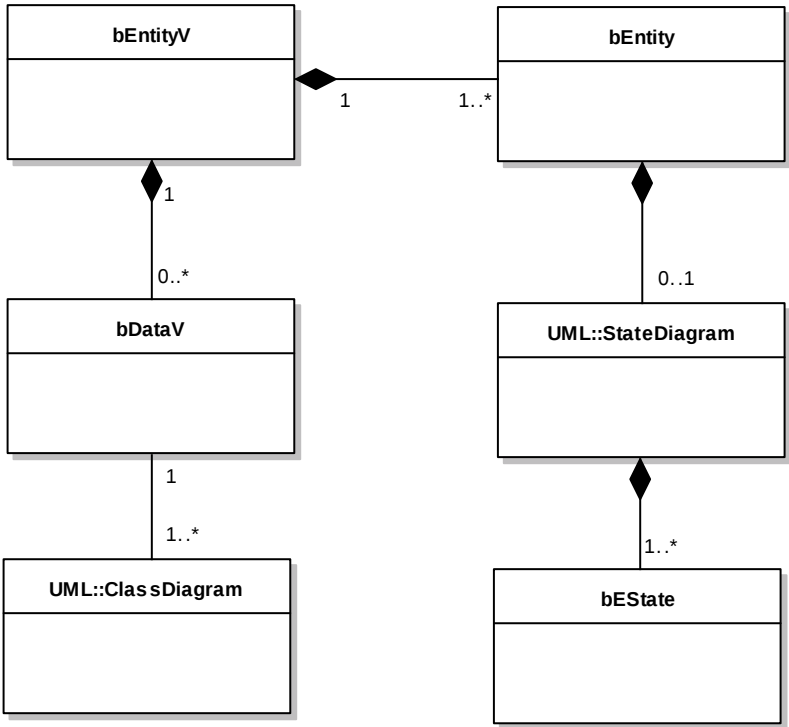
520 Figure 16 Business Partner View Example

521 5.1.3 Business Entity View

522 5.1.3.1 Abbreviations of Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bEntityV	BusinessEntityView
bEntity	BusinessEntity
bEState	BusinessEntityState
bDataV	BusinessDataView

523 5.1.3.2 Conceptual Description (informative)



524

525 Figure 17 BusinessEntityView (BusinessRequirementsView) Conceptual Overview

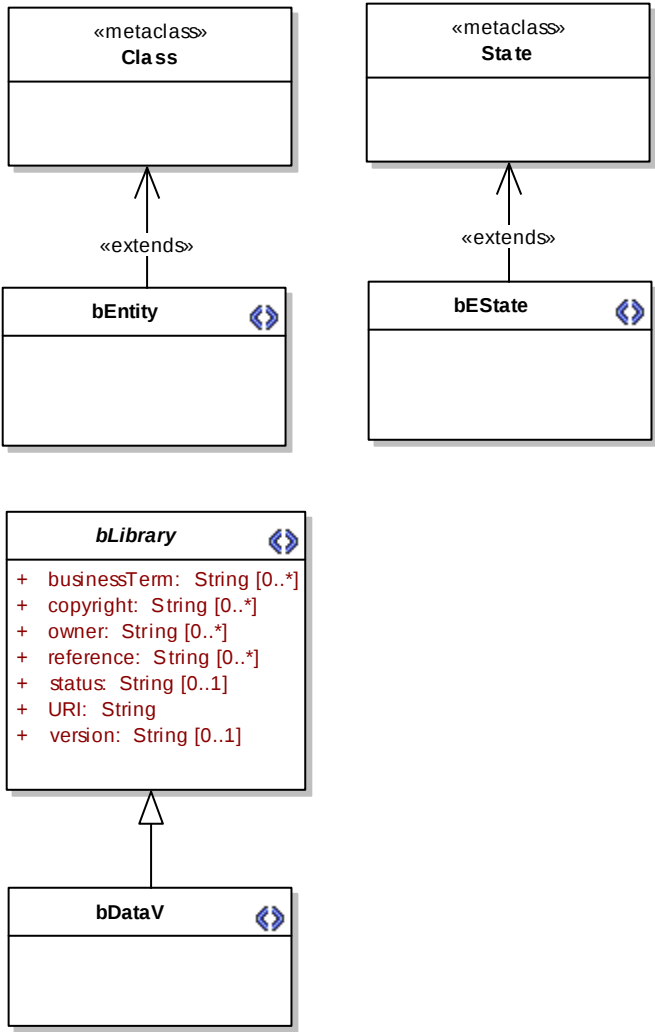
526

527 A business entity is a real-world thing having business significance that is shared between two or more
528 business partners in a collaborative business process (e.g. “order”, “account”, etc.). Within the business
529 entity view at least one, but possibly more business entities are described. Thus, the *BusinessEntityView* is
530 composed of one to many *BusinessEntities*. The lifecycle of a business entity MAY be described as a flow of
531 business entity states. Depending on the importance of the business entity lifecycle, the lifecycle may or
532 may not be included. A lifecycle is described using a UML State Diagram. Hence, a *BusinessEntity* is
533 composed of zero to one UML State Diagram. The lifecycle represents the different business entity states a
534 business entity can exist in. The lifecycle of a business entity consists of at least one business entity state.
535 Therefore, the lifecycle of a business entity is composed of one or more *BusinessEntityStates*.

536 A business entity is a potential candidate for becoming a business document in later steps of the UMM. A
537 business data view MAY be used to elaborate a first conceptual design of a business entity. Hence, a
538 *BusinessEntity* is composed of zero to one *BusinessDataViews*. Within a business data view, A UML class

539 diagram is used to describe the assembly of a business entity. Thus, a *BusinessDataView* contains one to
540 many UML Class Diagrams.

541 **5.1.3.3 Stereotypes and Tag Definitions (normative)**



542

543 **Figure 18 BusinessEntityView (BusinessRequirementsView) Abstract Syntax**

544

Stereotype	BusinessEntity
Base Class	Class
Parent	N/A
Description	A business entity is a real-world thing having business significance that is shared among two or more business partner types in a collaborative business process (e.g. order, account, etc.).
Tag Definition	No tagged values.

545

546

Stereotype	BusinessEntityState
Base Class	State
Parent	N/A
Description	A business entity state represents a specific state a business entity can exists in during its lifecycle (an “order” can exist in the states “issued”, “rejected”, “confirmed”, etc.)
Tag Definition	No tagged values.

547

Stereotype	BusinessDataView
Base Class	Package
Parent	BusinessLibrary
Description	The business data view is a container for all elements needed to describe the conceptual assembly of a business entity
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

548

5.1.3.4 Constraints (normative)

549

C.23. A *BusinessEntityView* MUST contain one to many *BusinessEntities*

550

C.24. A *BusinessEntity* SHOULD have zero to one UML State Diagram that describe its lifecycle

551

C.25. A UML State Diagram describing the lifecycle of a *BusinessEntity* MUST contain one to many

552

BusinessEntityStates

553

C.26. A *BusinessEntityView* MAY contain zero to many *BusinessDataView* that describe its conceptual design

554

555

C.27. The parent of a *BusinessDataView* MUST be a *BusinessEntityView*

556

C.28. A *BusinessDataView* SHOULD use a UML Class Diagram to describe the conceptual design of a

557

BusinessEntity

558

C.29. A *BusinessDataView* SHOULD contain one to many classes.

559

5.1.3.5 Worksheets

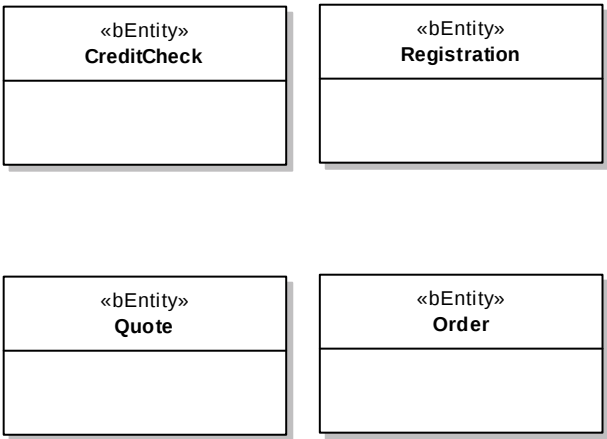
Form for Business Entity	
General	
Business Entity Name	
Description	

Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Lifecycle	
Pre-Condition	
Post-Condition	
Begins When	
Ends When	
Exceptions	
Lifecycle States (add more Business Entity States if needed)	
Business Entity State	
Name	
Description	
Preceding State(s) including events and transition conditions	
Valid Actions	
Business Entity State	
Name	
Description	
Preceding State(s) including events and transition conditions	

Valid Actions	
Business Entity State	
Name	
Description	
Preceding State(s) including events and transition conditions	
Valid Actions	
Business Entity State	
Name	
Description	
Preceding State(s) including events and transition conditions	
Valid Actions	
Business Entity State	
Name	
Description	
Preceding State(s) including events and transition conditions	
Valid Actions	

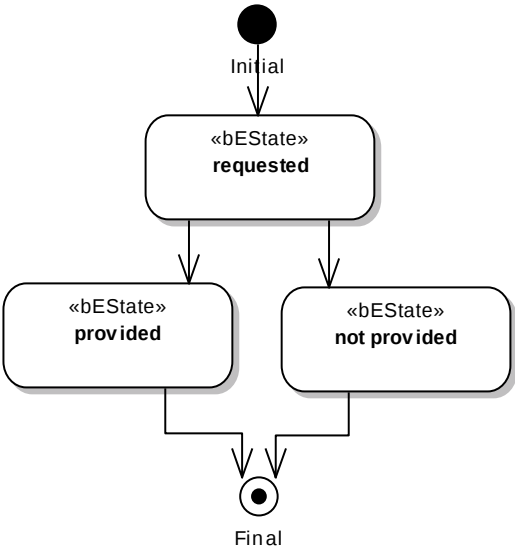
560

561 5.1.3.6 Example (informative)



562

563 Figure 19 BusinessEntityView Example: BusinessEntities - CreditCheck, Registration, Quote and Order (Class Diagram)



564

565 Figure 20 BusinessEntityView Example: CreditCeck Lifecycle (State Diagram)

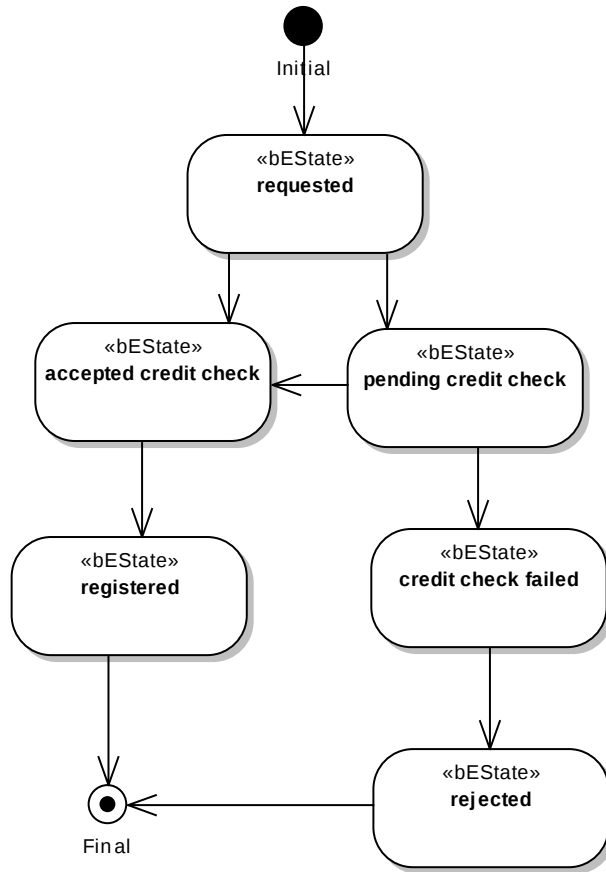


Figure 21 BusinessEntityView Example: Registration Lifecycle (State Diagram)

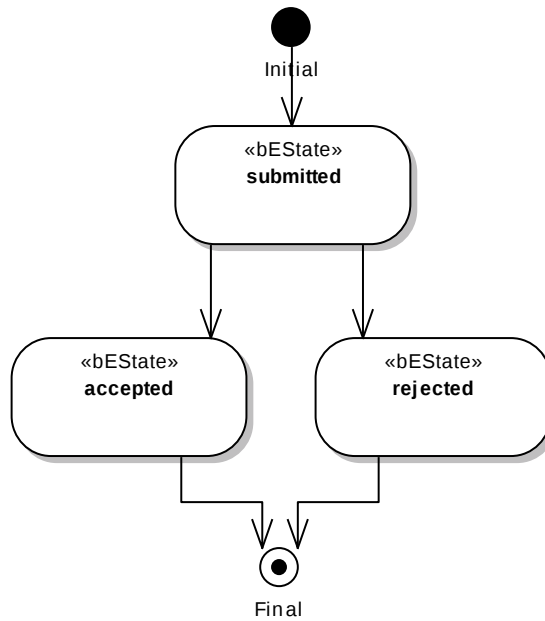


Figure 22 BusinessEntityView (BusinessRequirementsView) Example: Order BusinessEntityLifecycle (State Diagram)

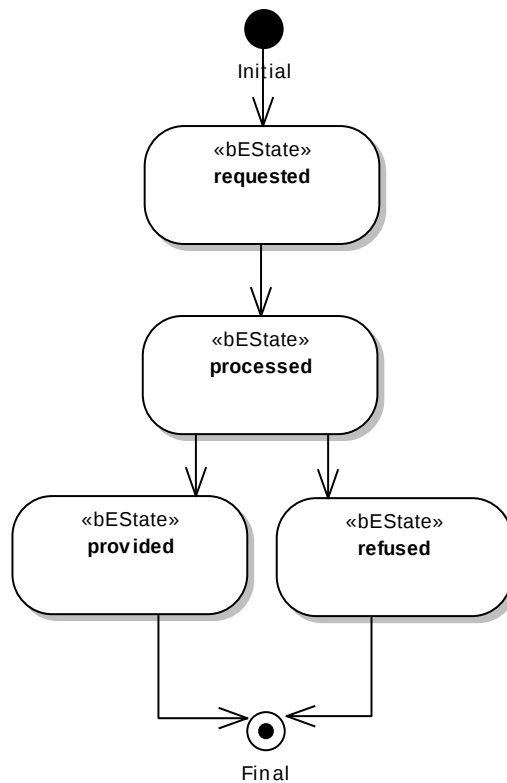


Figure 23 BusinessEntityView (BusinessRequirementsView) Example: Quote BusinessEntityLifecycle (State Diagram)

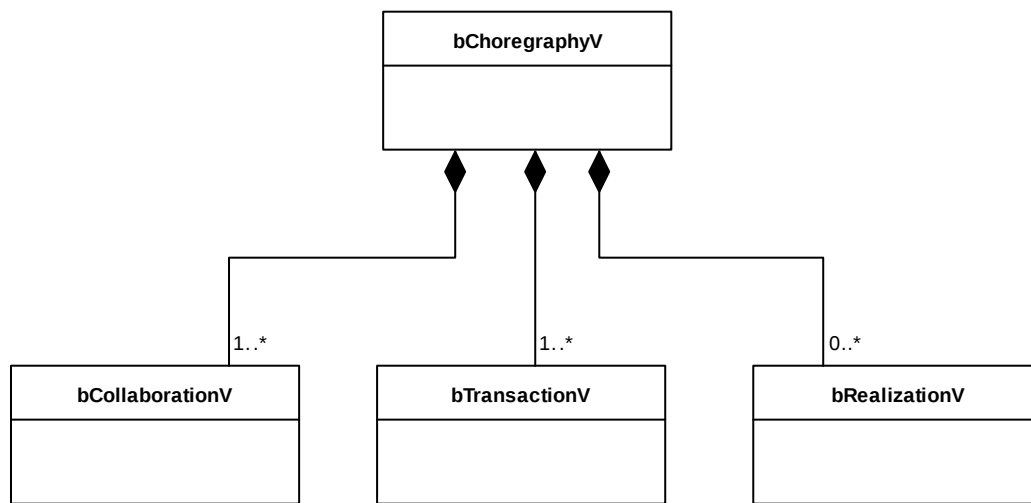
5.2 Business Choreography View

5.2.1 Sub-Views in the Business Choreography View

5.2.1.1 Abbreviations and Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bChoreographyV	BusinessChoreographyView
bCollaborationV	BusinessCollaborationView
bTransactionV	BusinessTransactionView
bRealizationV	BusinessRealizationView

578 **5.2.1.2 Conceptual Description (informative)**



579
580 **Figure 24 BusinessChoreographyView Conceptual Overview**

581 The *BusinessChoreographyView* is the second out of the 3 views of a UMM compliant business collaboration
 582 model. The business choreography view describes the view how the business analyst sees the process to be
 583 modeled. The requirements captured in the business requirements view serve as a basis for the definition of
 584 a choreography of information exchanges. The business choreography view is a container for three different
 585 artifacts that together describe the overall choreography of information exchanges. A
 586 *BusinessTransactionView* is a container for artifacts that define a choreography leading to synchronized
 587 states of business entities at both sides of the interaction. In fact, a business transaction view captures two
 588 different artifacts that define the business transaction. First, the business analyst defines concrete
 589 requirements specifying the business transaction on a more general level by using business transaction use
 590 cases. Second, he defines the flow of information exchanges in accordance to the requirements specified in
 591 this container. The business collaboration view is a container for artifacts describing the flow of a complex
 592 business collaboration between business partner types that may involve many steps. Similar to the business
 593 transaction view, the *BusinessCollaborationView* captures two different artifacts as well. Once the business
 594 analyst has specified the concrete requirements for a business collaboration by using business collaboration
 595 use cases, he is able to define the flow in accordance to the requirements defined in this container. Finally,
 596 the *CollaborationRealizationView* describes the realization of a business collaboration use case for a specific
 597 set of business partner types.

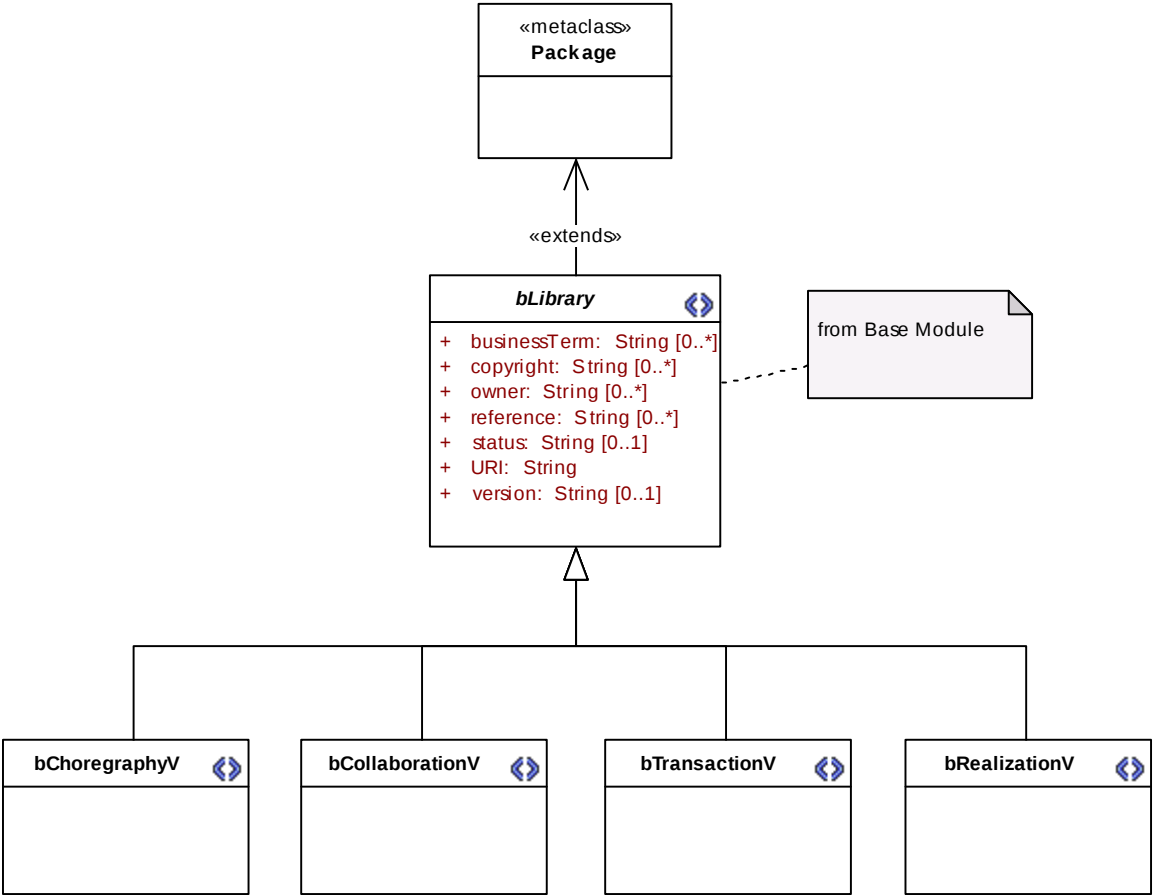


Figure 25 BusinessChoreographyView Abstract Syntax

Stereotype	BusinessChoreographyView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business choreography view is a container for all elements needed to describe the choreography of a business collaboration at various levels and the information exchanged in each step of the choreography.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

Stereotype	BusinessCollaborationView
------------	---------------------------

Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business collaboration view is a container for artifacts describing the flow of a complex business collaboration between business partner types that may involve many steps.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

603

Stereotype	BusinessTransactionView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The transaction requirements view is a container for artifacts that define a choreography leading to synchronized states of business entities at both sides of the business transaction.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

604

Stereotype	BusinessRealizationView
Base Class	Package
Parent	BusinessLibrary (from Base Module)
Description	The business realization view is a container for all elements describing the realization of a business collaboration use case by business partner types.
Tag Definition	Inherited tagged values: – URI – owner – copyright – reference – version – status – businessTerm

605

606

607 **5.2.1.4 Constraints (normative)**

608 Constraints with respect to a *BusinessChoreographyView*:

609 C.30. A *BusinessChoreographyView* MUST contain one to many *BusinessCollaborationViews*

610 C.31. A *BusinessChoreographyView* MUST contain one to many *BusinessTransactionViews*.

611 C.32. A *BusinessChoreographyView* MAY contain zero to many *BusinessRealizationViews*.

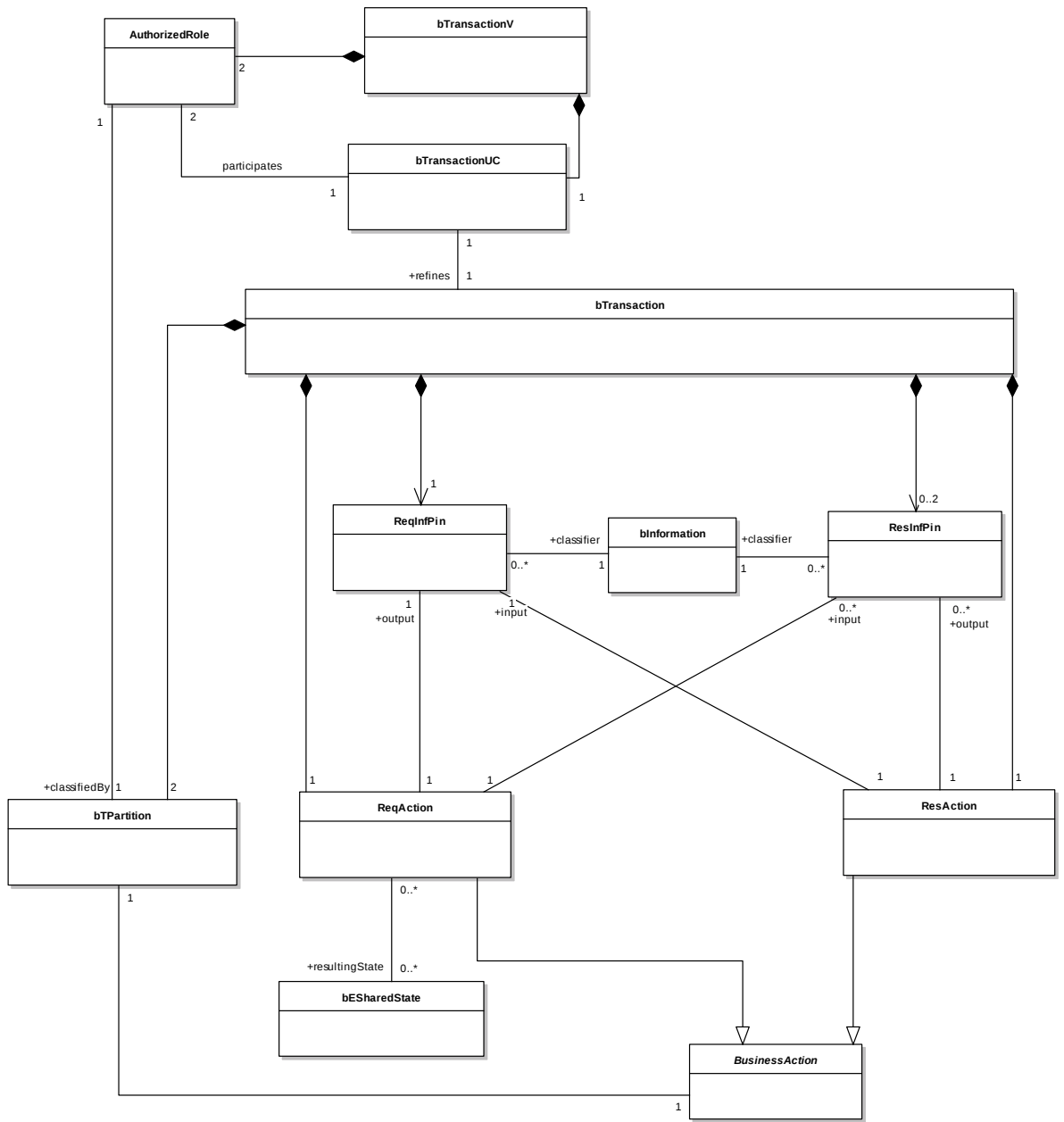
612 C.33. A *BusinessTransactionView*, a *BusinessCollaborationView*, and a *BusinessRealizationView* MUST be
613 directly located under a *BusinessChoreographyView*

614 **5.2.2 Business Transaction View**

615 **5.2.2.1 Abbreviations of Stereotypes**

Stereotype Abbreviation	Full Stereotype Name
bTransactionView	BusinessTransactionView
AuthorizedRole	AuthorizedRole
bTransactionUC	BusinessTransactionUseCase
bTransaction	BusinessTransaction
bTPartition	BusinessTransactionPartition
bInformation	BusinessInformation
ReqInfPin	RequestingInformationPin
ResInfPin	RespondingInformationPin
InfPin	InformationPin
BusinessAction	BusinessAction
ReqAction	RequestingBusinessAction
ResAction	RespondingBusinessAction
bESharedState	SharedBusinessEntityState

616



619
620 **Figure 26 Business Transaction View – Conceptual Overview**

621 Each *BusinessTransactionView* defines exactly one message exchange that leads to a synchronized business
622 state between the two authorized roles executing it. The flow of messages is specified by the concept of a
623 *BusinessTransaction*. The requirements of a *BusinessTransaction* are captured by a
624 *BusinessTransactionUseCase*.

625 Each business transaction and its corresponding business transaction use case are defined in their own
626 business transaction view package. Accordingly, the business transaction view is composed of exactly one
627 *BusinessTransactionUseCase* and one *BusinessTransaction*.

5.2.2.2.1 Business Transaction Use Case Diagram

Two authorized roles participate in a business transaction use case. These authorized roles must be defined in the same business transaction view package as the corresponding business transaction use case. Accordingly, a *BusinessTransactionView* is composed of exactly two *AuthorizedRoles*. This means, if a certain role (e.g. buyer, seller, etc.) participates in multiple business transactions, it requires a different authorized role for each business transaction use case. Each authorized role of the same role (i.e., with the same name) is in a different namespace of a corresponding business transaction view. Therefore, an authorized role participates in only one business transaction use case – it is the one in the same business transaction view. Accordingly, *BusinessTransactionUseCase* and *AuthorizedRole* are related by a 1 to 2 association. It is important to note, that the same authorized role is not associated twice to the same business transaction use case.

5.2.2.2.2 Business Transaction Diagram

A *BusinessTransaction* choreographs the synchronization of business states and the required information exchange between two authorized roles. The business transaction follows exactly the requirements defined in the corresponding business transaction use case. The business transaction that describes the business transaction use case is defined as a child beneath. Accordingly, each *BusinessTransactionUseCase* has exactly one *BusinessTransaction* beneath. A business transaction is a “composite” UML *Activity*. The graph of a business transaction is described by a flow of UML *Actions*.

A business transaction is an atomic step in a collaborative business process between two authorized roles, which involves sending business information from one authorized role to the other and an optional reply. The business transaction is built by two partitions - one for each authorized role. Hence, a *BusinessTransaction* is composed of exactly two *BusinessTransactionPartitions*. Each *BusinessTransactionPartition* relates to one *AuthorizedRole*. An *Authorized Role* is assigned to exactly one *BusinessTransactionPartition*. It follows, that the two partitions of a business transaction must be assigned to different authorized roles.

Within a business transaction each authorized role performs exactly one business action – the requesting authorized role performs a requesting business action and the responding authorized role performs a responding business action. Each business action – no matter whether requesting or responding business action – is assigned to a swimlane, and each swimlane comprises exactly one business action. It follows that a *BusinessTransaction* is composed of exactly one *RequestingBusinessAction* and exactly one *RespondingBusinessAction*. Both, *RequestingBusinessAction* and *RespondingBusinessAction* are specializations of the abstract type *BusinessAction*. A *BusinessAction* is assigned to one *BusinessTransactionPartition*, and a *BusinessTransactionPartition* comprises one *BusinessAction*. Since a partition is dedicated to exactly one authorized role, it follows that the business action is executed by this authorized role. Furthermore an authorized role executes just one business action, because only one business action sits within a partition.

The requesting business action outputs the requesting information through the requesting information pin that is input to the responding business action’s requesting information pin. Business information created by the responding business action and returned to the requesting business action is optional. If business information is returned by the responding business action zero to many responding information pins might be specified. Multiple responding information pins may be used to describe different business intentions (e.g., a positive and a negative response to a purchase order).

670 It follows, that a *BusinessTransaction* is composed of exactly two *RequestingInformationPins* and zero to
671 many *RespondingInformationPins*. Both *RequestingInformationPin* and *RespondingInformationPin* are
672 instances of the type *BusinessInformation*. A *RequestingBusinessAction* has exactly one
673 *RequestingInformationPin* and zero to many *RespondingInformationPins*.

674 A *RespondingBusinessAction* has exactly one *RequestingInformationPin* and zero to many
675 *RespondingInformationPins*.

676 *RequestingInformationPin* and *RespondingInformationPin* are stereotypes of the UML base class *Pin*. The
677 type of the *Pin* is defined by the *BusinessInformation* that is a stereotype of the UML base class *Class*.
678 According to UML, multiple *Pins* might be instances of the same *Class*. It follows that different requesting or
679 responding information pins might be instances of the same business information. In other words, business
680 information might be reused in different business transactions. Action pins that specify output information
681 from a business action MUST be stereotyped and classified accordingly, whereas action pins that specify
682 input information to a business action MAY not be stereotyped and classified.

683 If multiple responding information pins are defined, those must be in an XOR relationship with each other.
684 In order to specify an XOR relationship between multiple incoming or outgoing responding information pins,
685 each of them has to be enclosed by an UML *ParameterSet* (c.f. Figure 32). If only one responding information
686 pin is defined within a business transaction, *ParameterSets* SHOULD not be used.

687 In order to determine the outcome of a business transaction (success or failure) the contents of the
688 responding business document SHOULD be evaluated. OCL constraints SHOULD be used for assessing the
689 document's content. An OCL constraint may either check the responding business information's type (e.g.,
690 positive or negative response to a quote) or directly investigate the document's content (e.g., if products
691 were quoted or not). If the responding business information is checked, the constraints MUST be applied as
692 condition guards to the transitions leading into the respective final states (e.g., success or response) of the
693 business transaction. If the business transaction does not include a response, OCL constraints MAY not be
694 used.

695 A business transaction synchronizes the states between the two authorized roles executing it. Thus, the
696 execution of a business transaction results in a certain business entity shared state (the concept of business
697 entity shared states have already been introduced in X). In order to point out the state change, setting the
698 resulting shared state of a business entity might be visualized on the diagram of a business transaction. A
699 *SharedBusinessEntityState* MAY be included as a predecessor of a final state to indicate the resulting
700 synchronized state. The example in Figure 33 illustrates this concept.

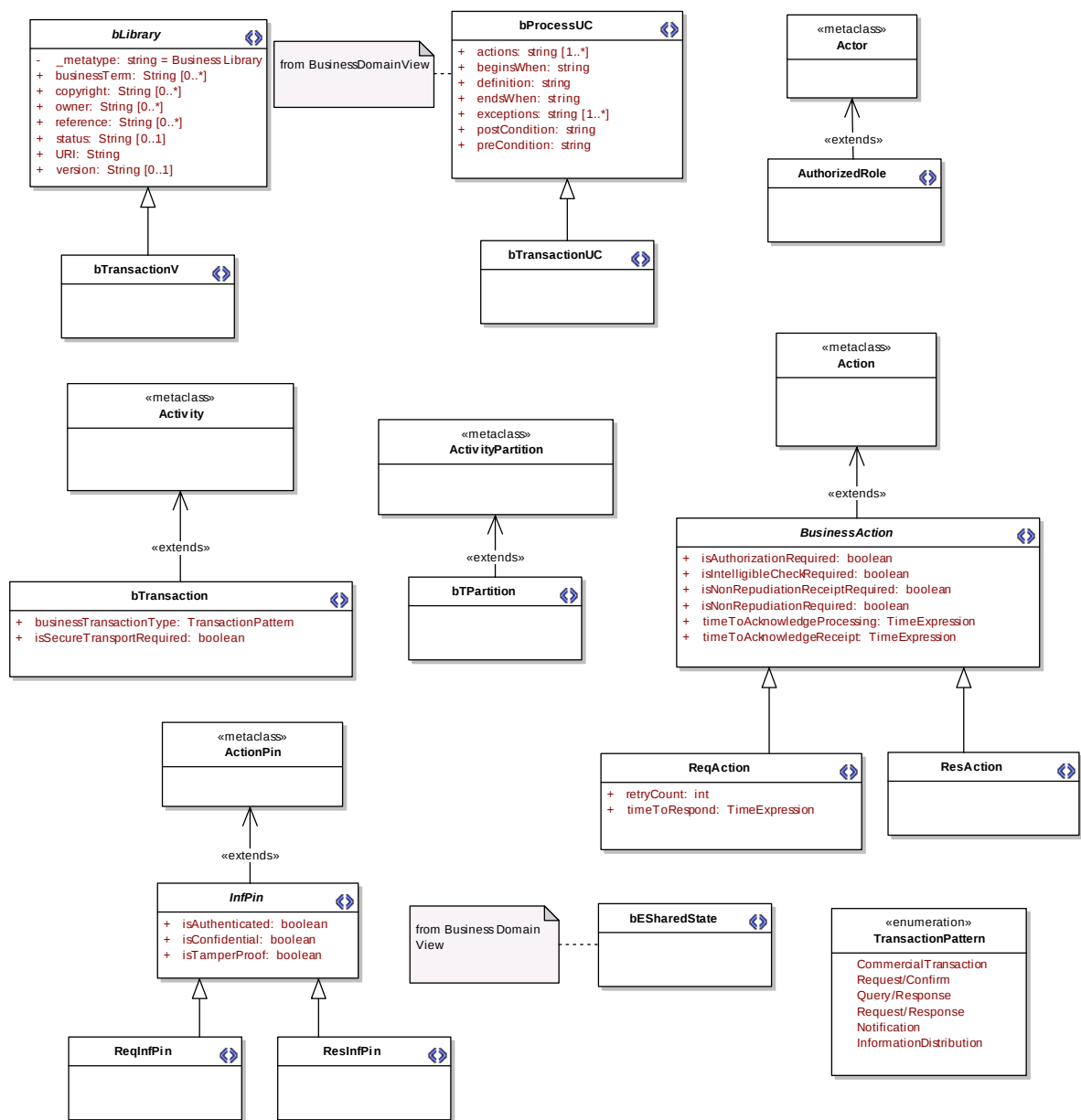


Figure 27 Business Transaction View – Abstract Syntax

Stereotype	bTransactionUC (BusinessTransactionUseCase)
Base Class	UseCase
Parent	bProcessUC
Description	A business transaction use case describes in detail the requirements on a collaboration between exactly two involved partners. A business transaction use case cannot be further refined and describes the requirements on a one-way or two-way information exchange. Business partners take part in a business transaction use case by playing an authorized role in it.
Tag Definition	Inherited tagged values: - definition - beginsWhen

704

	- preCondition - endsWhen - postCondition - exceptions - actions
--	--

705

Stereotype	AuthorizedRole (AuthorizedRole)
Base Class	Actor
Parent	N/A
Description	An authorized role (e.g. a “buyer”) is a concept which is more generic than a business partner (e.g. a “wholesaler”) and allows the reuse of collaborations by mapping an <i>AuthorizedRole</i> to a business partner within a given scenario. Since business collaboration use case and business transaction use case are defined as occurring between authorized roles, they might be reused by different business partners (a “wholesaler” or a “broker”) in different scenarios of the same domain or even in different domains.
Tag Definition	No tagged values.

Stereotype	bTransaction (BusinessTransaction)
Base Class	Activity
Parent	N/A
Description	A business transaction is the basic building block to define choreography between authorized roles. If an authorized role recognizes an event that changes the state of a business object, it initiates a business transaction to synchronize with the collaborating authorized role. It follows that a business transaction is an atomic unit that leads to a synchronized state in both information systems. We distinguish one-way and two-way business transaction: In the former case, the initiating authorized role reports an already effective and irreversible state change that the reacting authorized role has to accept. Examples are the notification of shipment or the update of a product in a catalog. It is a one-way business transaction, because business information (not including business signals for acknowledgments) flows only from the initiating to the reacting authorized role. In the other case, the initiating partner sets the business object(s) into an interim state and the final state is decided by the reacting authorized role. Examples include request for registration, search for products, etc. It is a two-way transaction, because business information flows from the initiator to the responder to set the interim state and backwards to set the final and irreversible state change. In a business context irreversible means that returning to an original state requires another – compensating – business transaction. E.g., once a purchase order is agreed upon in a business transaction a rollback is not allowed anymore, but requires the execution of a cancel order business transaction compensating the before sent purchase order. We distinguish 2 one-way business transactions and four two-way business transactions. The type of transaction is indicated in the tagged value of business transaction type. The other tagged values provide quality of service parameters. A business transaction follows always the same pattern: A business transaction is performed between two authorized roles that are assigned to exactly one swimlane each. Each authorized role performs exactly one activity. An object flow between the requesting and the responding business activity is mandatory. Up to two object flows in the reverse direction are optional. If two object flows are defined as response, those are in an XOR relationship with each other (e.g., a purchase order is accepted or declined). According to the business transaction semantics, the requesting business activity does not end

	after sending the business information - it is still alive. The responding business activity may output the response which is returned to the still living requesting business activity.	
Tag Definition	businessTransactionType	
	Type	Enumeration: • Commercial Transaction • Request/Confirm • Query/Response • Request/Response • Notification • Information Distribution
	Multiplicity	1
	Description	The business transaction type determines a corresponding business transaction pattern. A business transaction pattern provides a language and grammar for constructing business transactions. The business transaction type follows one of the following six property-value conventions: (1) Commercial Transaction - 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 (2) Query/Response – used to query for information that a responding partner already has e.g. against a fixed data set that resides in a database (3) Request/Response - used for business contracts when an initiating partner requests information that a responding partner already has and when the request for business information requires a complex interdependent set of results (4) Request/Confirm - used if an initiating partner asks for information that requires only confirmation with respect to previously established contracts or with respect to a responding partner’s business rules (5) Information Distribution - used to model an informal information exchange business transaction that therefore has no non-repudiation requirements (6) Notification - used to model a formal information exchange business transaction that therefore has non-repudiation requirements
	isSecureTransportRequired	
	Type	Boolean
	Multiplicity	1
	Description	Both partners must agree to exchange business information using a secure transport channel. The following security controls ensure that business document content is protected against unauthorized disclosure or modification and that business services are protected against unauthorized access. This is a point-to-point security requirement. Note that this requirement does not protect business information once it is off the network and inside an enterprise. The following are requirements for secure transport channels. Authenticate sender identity – Verify the identity of the sender (employee or organization) that is initiating the interaction (authenticate). For example, a driver’s license or passport document with a picture is used to verify an individual’s identity

706

		by comparing the individual against the picture. Authenticate receiver identity – Verify the identity of the receiver (employee or organization) that is receiving the interaction. Verify content integrity – Verify the integrity of the content exchanged during the interaction i.e. check that the content has not been altered by a 3rd party. Maintain content confidentiality – Confidentiality ensures that only the intended, receiver can read the content of the interaction. Information exchanged during the interaction must be encrypted when sent and decrypted when received. For example, you seal envelopes so that only the recipient can read the content.
--	--	--

707

Stereotype	bPartition (BusinessTransactionPartition)
Base Class	Partition
Parent	N/A
Description	A business transaction partition is used to define an area of responsibility. An authorized role is appointed to a business transaction swimlane to indicate that this authorized role takes on the responsibility for the business action that is allocated within that area.
Tag Definition	No Tagged Values

Stereotype	BusinessAction (BusinessAction, abstract)	
Base Class	Action	
Parent	N/A	
Description	A business action is executed by an authorized role during a business transaction. Business action is an abstract stereotype. This means a business action is either a requesting business action or a responding business action.	
Tag Definition	isAuthorizationRequired	
	Type	Boolean
	Multiplicity	1
	Description	If an authorized role needs authorization to request a business action or to respond to a business action then the sender must sign the business document exchanged and the receiver must validate this business control and approve the authorizer. A receiver must signal an authorization exception if the sender is not authorized to perform the business activity. A sender must send notification of failed authorization if a receiver is not authorized to perform the responding business activity.
	isNonRepudiationRequired	
	Type	Boolean
	Multiplicity	1
	Description	The <i>isNonRepudiationRequired</i> tag is used to indicate that an involved party must not be able to repudiate the execution of the business action that input/outputs

		business information.
	isNonRepudiationReceiptRequired	
	Type	Boolean
	Multiplicity	1
	Description	The <i>isNonRepudiationOfReceiptRequired</i> tag requires the receiver of a business information to send a signed receipt. The <i>isNonRepudiationOfReceiptRequired</i> tag indicates that an involved party must not be able to repudiate the execution of sending the signed receipt.
	timeToAcknowledge Receipt	
	Type	TimeExpression
	Multiplicity	1
	Description	Both partners may agree to mutually verify receipt of business information within a specific time duration. Acknowledgements of receipt may be sent for both the requesting business information and the responding business information. This means the sender of the business information may be the requesting authorized role as well as the responding authorized role – it depends on whether a requesting or a responding business information is acknowledged. Similarly, the affirmant may be the requesting authorized role as well as the responding authorized role – again depending of which business information is acknowledged. Inasmuch we use the terms sender and affirmant in the explanation of acknowledgement of receipt semantics. An affirmant must exit the transaction if they are not able to verify the proper receipt of a business information within the agree timeout period. A sender must retry a business transaction if necessary or must send notification of failed business control (possibly revoking a contractual offer) if an affirmant does not verify properly receipt of a business information within the agreed time period. The time to acknowledge receipt is the maximum duration from the time a business information is sent by a sender until the time a verification of receipt is “properly received” by the sender (of the business information). Accordingly, the time to acknowledge receipt is always specified by the sender’s business action. This verification of receipt is an audit-able business signal and is instrumental in contractual obligation transfer during a contract formation process (e.g. offer/accept).
	timeToAcknowledgeProcessing	
	Type	TimeExpression
	Multiplicity	1
	Description	Similarly to the <i>timeToAcknowledgeReceipt</i>, the sender of a business information might be the requesting authorized role as well as the responding authorized role – depending whether a requesting or a responding business information is acknowledged. Also the affirmant may be one of the two authorized roles. Thus, we use again the terms sender and affirmant in the explanation of the acknowledgment of processing semantics. Both partners may agree to the need for an acknowledgment of processing to be

		returned by a responding partner after the requesting business information passes a set of business rules and is handed over to the application for processing. The time to acknowledge processing of a business information is the duration from the time a sender sends a business information until the time an acknowledgement of processing is “properly received” by the sender (of the business information). Accordingly, the time to acknowledge processing is always specified by the sender’s business action. An affirmant must exit the transaction if they are not able to acknowledge processing of business information within the maximum timeout period. A sender must retry a business transaction if necessary or must send notification of failed business control (possibly revoking a contractual offer) if an affirmant does not acknowledge processing of business information within the agreed time period.
	isIntelligibleCheckRequired	
	Type	Boolean
	Multiplicity	1
	Description	In order to define the <i>isIntelligibleCheckRequired</i> semantics, we use again the terms sender and affirmant as introduced for the last two tag definitions. Both partners may agree that an affirmant must check that business information is not garbled (unreadable, unintelligible) before verification of proper receipt is returned to the sender (of the business information). Verification of receipt must be returned when a document is “accessible” but it is preferable to also check for garbled transmissions at the same time in a point-to-point synchronous business network where partners interact without going through an asynchronous service provider.

Stereotype	ReqAction (RequestingBusinessAction)	
Base Class	Action	
Parent	BusinessAction	
Description	A requesting business action is a business action that is performed by an authorized role requesting business service from another authorized role.	
Tag Definition	timeToRespond	
	Type	TimeExpression
	Multiplicity	1
	Description	Both partners may agree in case of a two-way business transaction that the responding authorized role must return the responding information business information within a specific duration. A responding authorized role must exit the transaction if they are not able to return the responding business information within the agreed timeout period. A requesting authorized role must retry a business transaction if necessary or must send notification of failed business control (possibly revoking a contractual offer) if a responding authorized role does not deliver the responding business information within the agreed time period. The time to perform is the maximum duration from

	the time a requesting business information is sent by a requesting authorized role until the time a responding business information is “properly received” by the requesting authorized role in return.																																																															
retryCount																																																																
Type	Integer																																																															
Multiplicity	1																																																															
Description	The requesting authorized role must re-initiate the business transaction so many times as specified by the retry count in case that a time-out-exception – by exceeding the time to acknowledge receipt, or the time to acknowledge processing, or the time to respond – is signaled. This parameter only applies to time-out signals and not document content exceptions or sequence validation exceptions – i.e., failed business control exceptions.																																																															
Inherited tagged values: - isAuthorizationRequired - isNonRepudiationRequired - isNonRepudiationReceiptRequired - timeToAcknowledgeReceipt - timeToAcknowledgeProcessing - isIntelligibleCheckRequired																																																																
Default assignment of tagged values for the requesting business action:																																																																
	<table><tr><td></td><td>Time to Acknowledge Receipt</td><td>Time to Acknowledge Processing</td><td>Time to Respond</td><td>Is Authorization Required</td><td>Is Non Repudiation Required</td><td>Is Non Repudiation of Receipt Required</td><td>Retry Count</td><td>Is Intelligible Check Required</td></tr><tr><td>Commercial Transaction</td><td>2h</td><td>6h</td><td>24h</td><td>TRUE</td><td>TRUE</td><td>TRUE</td><td>3</td><td>TRUE</td></tr><tr><td>Request/Confirm</td><td>NULL</td><td>NULL</td><td>24h</td><td>FALSE</td><td>FALSE</td><td>FALSE</td><td>3</td><td>TRUE</td></tr><tr><td>Request/Response</td><td>NULL</td><td>NULL</td><td>4h</td><td>FALSE</td><td>FALSE</td><td>FALSE</td><td>3</td><td>TRUE</td></tr><tr><td>Query/Response</td><td>NULL</td><td>NULL</td><td>4h</td><td>FALSE</td><td>FALSE</td><td>FALSE</td><td>3</td><td>TRUE</td></tr><tr><td>Notification</td><td>24h</td><td>NULL</td><td>NULL</td><td>FALSE</td><td>TRUE</td><td>TRUE</td><td>3</td><td>TRUE</td></tr><tr><td>Information Distribution</td><td>NULL</td><td>NULL</td><td>NULL</td><td>FALSE</td><td>FALSE</td><td>FALSE</td><td>0</td><td>TRUE</td></tr></table>		Time to Acknowledge Receipt	Time to Acknowledge Processing	Time to Respond	Is Authorization Required	Is Non Repudiation Required	Is Non Repudiation of Receipt Required	Retry Count	Is Intelligible Check Required	Commercial Transaction	2h	6h	24h	TRUE	TRUE	TRUE	3	TRUE	Request/Confirm	NULL	NULL	24h	FALSE	FALSE	FALSE	3	TRUE	Request/Response	NULL	NULL	4h	FALSE	FALSE	FALSE	3	TRUE	Query/Response	NULL	NULL	4h	FALSE	FALSE	FALSE	3	TRUE	Notification	24h	NULL	NULL	FALSE	TRUE	TRUE	3	TRUE	Information Distribution	NULL	NULL	NULL	FALSE	FALSE	FALSE	0	TRUE
	Time to Acknowledge Receipt	Time to Acknowledge Processing	Time to Respond	Is Authorization Required	Is Non Repudiation Required	Is Non Repudiation of Receipt Required	Retry Count	Is Intelligible Check Required																																																								
Commercial Transaction	2h	6h	24h	TRUE	TRUE	TRUE	3	TRUE																																																								
Request/Confirm	NULL	NULL	24h	FALSE	FALSE	FALSE	3	TRUE																																																								
Request/Response	NULL	NULL	4h	FALSE	FALSE	FALSE	3	TRUE																																																								
Query/Response	NULL	NULL	4h	FALSE	FALSE	FALSE	3	TRUE																																																								
Notification	24h	NULL	NULL	FALSE	TRUE	TRUE	3	TRUE																																																								
Information Distribution	NULL	NULL	NULL	FALSE	FALSE	FALSE	0	TRUE																																																								

709

710

Stereotype	ResAction (RespondingBusinessAction)						
Base Class	ActionState						
Parent	Business Action						
Description	A responding business activity is a business action that is performed by an authorized role responding to another authorized role’s request for business service.						
Tag Definition	Inherited tagged values: - isAuthorizationRequired - isNonRepudiationRequired - isNonRepudiationReceiptRequired - timeToAcknowledgeReceipt - timeToAcknowledgeProcessing - isIntelligibleCheckRequired Default assignment of tagged values for the requesting business action:						
		Time to Acknowledge Receipt	Time to Acknowledge Processing	Is Authorization Required	Is Non Repudiation Required	Is Non Repudiation of Receipt Required	Is Intelligible Check Required
	Commercial Transaction	2h	6hr	TRUE	TRUE	TRUE	TRUE
	Request/Confirm	2h	NULL	TRUE	FALSE	TRUE	TRUE
	Request/Response	NULL	NULL	FALSE	FALSE	FALSE	TRUE
	Query/Response	NULL	NULL	FALSE	FALSE	FALSE	TRUE
	Notification	NULL	NULL	FALSE	FALSE	FALSE	TRUE
	Information Distribution	NULL	NULL	FALSE	FALSE	FALSE	TRUE

711

Stereotype	InfPin (InformationPin, abstract)
Base Class	Pin
Parent	N/A
Description	The abstract concept information pin represents the incoming/outgoing point for business information in a business action. Business information is sent from the requesting authorized role to the responding authorized role or the reverse way. The actual exchanged information is represented using the type business information. Both concrete stereotypes requesting information pin and responding information

	pin inherit from the abstract stereotype information pin.	
Tag Definition	isConfidential	
	Type	Boolean
	Multiplicity	1
	Description	If the flag is set, the exchanged information is encrypted so that unauthorized parties cannot view the information.
	isTamperProof	
	Type	Boolean
	Multiplicity	1
	Description	If the flag is set, the exchanged information has an encrypted message digest that can be used to check if the message has been tampered with. This requires a digital signature (sender's digital certificate and encrypted message digest) associated with the document entity.
	isAuthenticated	
	Type	Boolean
	Multiplicity	1
	Description	If the flag is set, there is a digital certificate associated with the document entity. This provides proof of the signer's identity.

712

Stereotype	ReqInfPin (RequestingInformationPin)
Base Class	Pin
Parent	InformationPin
Description	The requesting information pin is a container for business information that is sent from the requesting authorized role to the responding authorized role to indicate a state change in one or more business entities. This business state change might be irreversible in the case of a one-way business transaction or an interim state of a two-way business transaction. It is important to note that the term requesting information pin does not mean that the exchanged business information refers to a request in a business sense. The term requesting information pin indicates that the execution of a transaction is requested from the requesting authorized role to the responding authorized role – no matter whether this is an information distribution, a notification, a request, or the offer in a commercial transaction.
Tag Definition	Inherited tagged values: - isConfidential - isAuthenticated - isTamperProof

713

Stereotype	ResInfPin (RespondingInformationPin)
Base Class	Pin
Parent	InformationPin

Description	The responding information pin is a container of business information that is sent in case of a two-way business transaction from the responding authorized role to the requesting authorized role in order to set one or more business entities in a final state (which were in an interim state before).
Tag Definition	Inherited tagged values: - isConfidential - isAuthenticated - isTamperProof

714

715 5.2.2.4 Constraints (normative)

- 716 C.34. A *BusinessTransactionView* MUST contain exactly one *BusinessTransactionUseCase*, exactly
717 two *AuthorizedRoles*, and exactly two *participates* associations.
- 718 C.35. A *BusinessTransactionUseCase* MUST be associated with exactly two *AuthorizedRoles* via
719 stereotyped binary *participates* associations.
- 720 C.36. A *BusinessTransactionUseCase* MUST NOT include further *UseCases*
- 721 C.37. A *BusinessTransactionUseCase* MUST be included in at least one
722 *BusinessCollaborationUseCase*.
- 723 C.38. A *BusinessTransactionUseCase* MUST NOT be source or target of an extend association.
- 724 C.39. The two *AuthorizedRoles* within a *BusinessTransactionView* MUST NOT be named identically
- 725 C.40. A *BusinessTransactionUseCase* MUST be described by exactly one *BusinessTransaction* defined as a
726 child element of this *BusinessTransactionUseCase*.
- 727 C.41. A *BusinessTransaction* MUST have exactly two partitions. Each of them MUST be stereotyped as
728 *BusinessTransactionPartition*.
- 729 C.42. One of the two *BusinessTransactionPartitions* MUST contain one *RequestingBusinessAction* and the
730 other one MUST contain one *RespondingBusinessAction*.
- 731 C.43. A *BusinessTransactionPartition* MUST have a classifier, which MUST be one of the associated
732 *AuthorizedRoles* of the corresponding *BusinessTransactionUseCase*.
- 733 C.44. The two *BusinessTransactionPartitions* MUST have different classifiers.
- 734 C.45. The *BusinessTransactionPartition* containing the *RequestingBusinessAction* MUST contain two or
735 more *FinalStates*. Each of the *FinalStates* MAY have a *BusinessEntitySharedState* as predecessor. One
736 of the *FinalStates* SHOULD reflect a *ControlFailure* – this *FinalState* SHOULD NOT have a
737 predeccessing *SharedBusinessEntityState*.
- 738 C.46. A *RequestingBusinessAction* MUST embed exactly one *RequestingInformationPin*
- 739 C.47. A *RespondingBusinessAction* MUST embed exactly one *RequestingInformationPin*
- 740 C.48. If the tagged value *businessTransactionType* of the *BusinessTransaction* is either *Request/Response*,
741 *Query/Response*, *Request/Confirm*, or *CommercialTransaction*, then the *RequestingBusinessAction*
742 must embed one to many *RespondingInformationPins* and the *RespondingBusinessAction* must
743 embed one to many *RespondingInformationPins*.
- 744 C.49. If the tagged value *businessTransactionType* of the *BusinessTransaction* is either *Notification* or
745 *InformationDistribution*, then both, the *RequestingBusinessAction* and the
746 *RespondingBusinessAction*, MUST NOT embed a *RespondingInformationPin*
- 747 C.50. A *RequestingBusinessAction* and a *RespondingBusinessAction* MUST embed same number of
748 *RespondingInformationPins*.

C.51. The *RequestingInformationPin* of the *RequestingBusinessAction* MUST be connected with the *RequestingInformationPin* of the *RespondingBusinessAction* using an object flow relationship leading from the *RequestingBusinessAction* to the *RespondingBusinessAction*.

C.52. Each *RespondingInformationPin* of the *RespondingBusinessAction* MUST be connected with exactly one *RespondingInformationPin* of the *RequestingBusinessAction* using an object flow relationship leading from the *RespondingBusinessAction* to the *RequestingBusinessAction*.

C.53. If a *BusinessTransactionPartition* contains *SharedBusinessEntityStates*, each *SharedBusinessEntityState* MUST be the target of exactly one control flow relationship starting from the *RequestingBusinessAction* and MUST be the source of exactly one control flow relationship targeting a *FinalState*.

C.54. Each *FinalState* MUST be the target of one to many control flow relationships starting from the *RequestingBusinessAction* or from a *SharedBusinessEntityState*.

C.55. Each *RequestingInformationPin* and each *RespondingInformationPin* MUST have a classifier, this classifier MUST be an *InformationEnvelope* or a subtype defined in an extension/specialization module.

C.56. Two *RequestingInformationPins* which are connected using an object flow MUST have the same classifier.

C.57. Two *RespondingInformationPins* which are connected using an object flow MUST have the same classifier.

5.2.2.5 Worksheets

Form for Business Transaction Use Case	
General	
Name	
Description	
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Details	
Requesting Role	

Responding Role	
Requesting Activity	
Responding Activity	
Is Included In (Name of Business Collaboration)	
Start / End Characteristics	
Affected Business Entities	
Pre-condition	
Post-condition	
Begins When	
Ends When	
Exceptions	

769

Form for Business Transaction	
General	
Name	
Description	
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Details	

Select Business Transaction Pattern	☐ Information Distribution ☐ Notification ☐ RequestResponse ☐ RequestConfirm ☒ QueryResponse ☐ Commercial Transaction
Secure Transport	
Requestor's Side	
Requesting Role	
Requesting Business Action Name	
Time to Respond	
Time to Acknowledge Receipt	
Time to Acknowledge Processing	
Authorization Required	
Non Repudiation Required	
Non Repudiation of Receipt Required	
Intelligible Check Required	
Number of Retries	
Responder's Side	
Responding Role	
Responding Business Action Name	
Time to Acknowledge Receipt	
Time to Acknowledge Processing	

Authorization Required	
Non Repudiation Required	
Non Repudiation of Receipt Required	
Intelligible Check Required	
Business Information Envelopes	
Requesting Information Envelope	
Name	
Are Contents Confidential?	
Is the Envelope Tamperproof?	
Authentication Required?	
Responding Information Envelope (add more Responding Information Envelopes if different response documents are possible)	
Name	
Resulting Business Entity State (including transition condition)	
Are Contents Confidential?	
Is the Envelope Tamperproof?	
Authentication Required?	

770

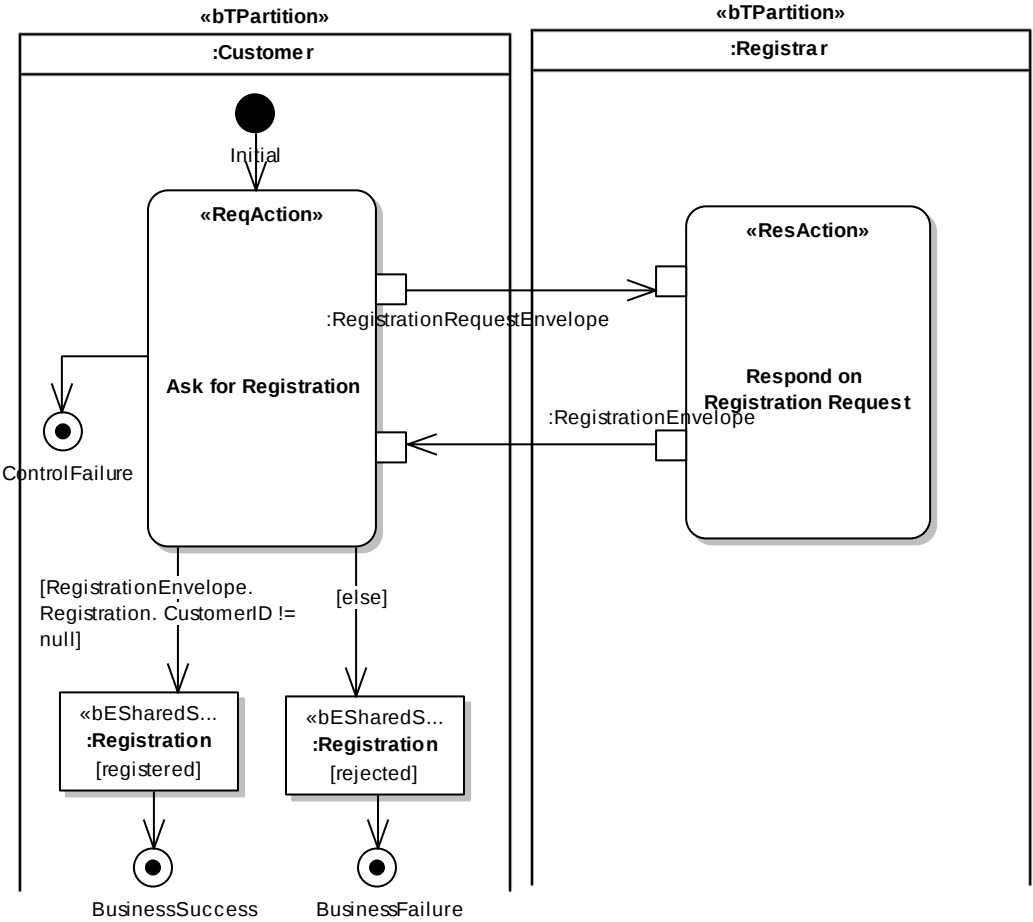
771 5.2.2.6 Example (informative)



772

773

Figure 28 Business Transaction Use Case Example: Register Customer



774

775

776

Figure 29 Business Transaction Example: Register Customer



777

778

Figure 30 Business Transaction Use Case Example: Request For Quote

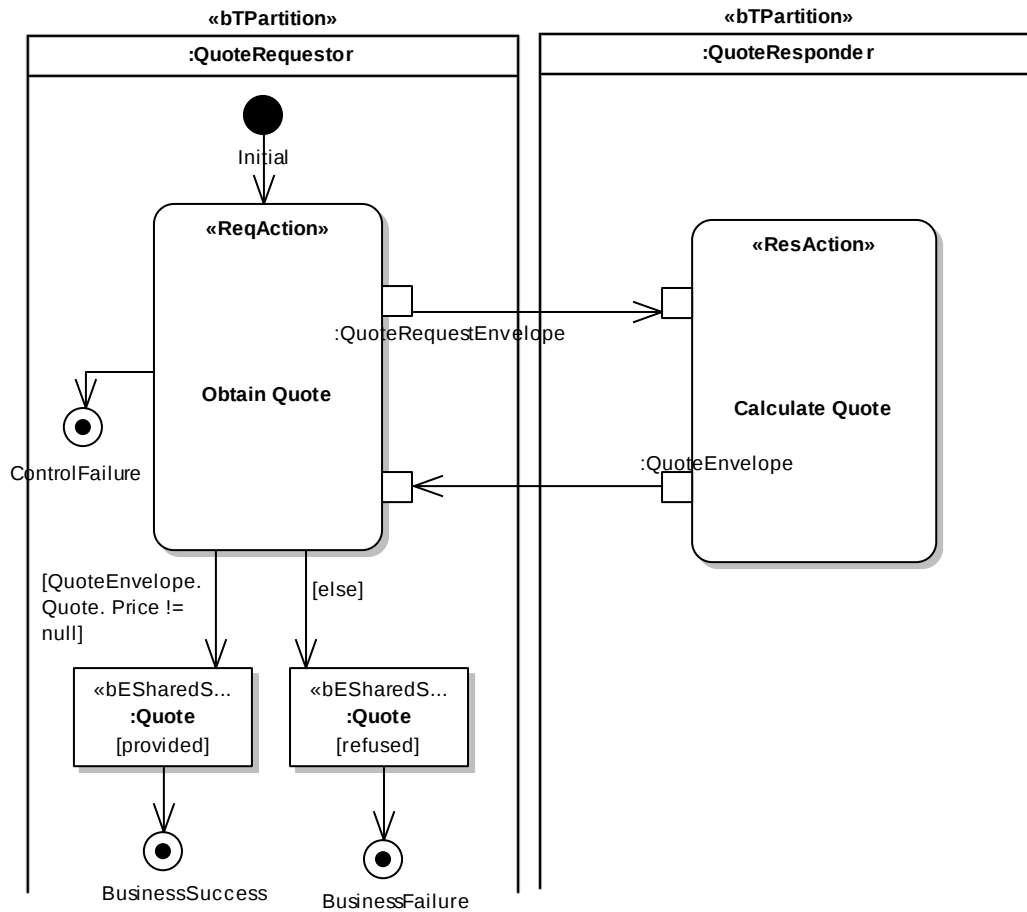


Figure 31 Business Transaction Example: Request For Quote



Figure 32 Business Transaction Use Case Example: Place Order

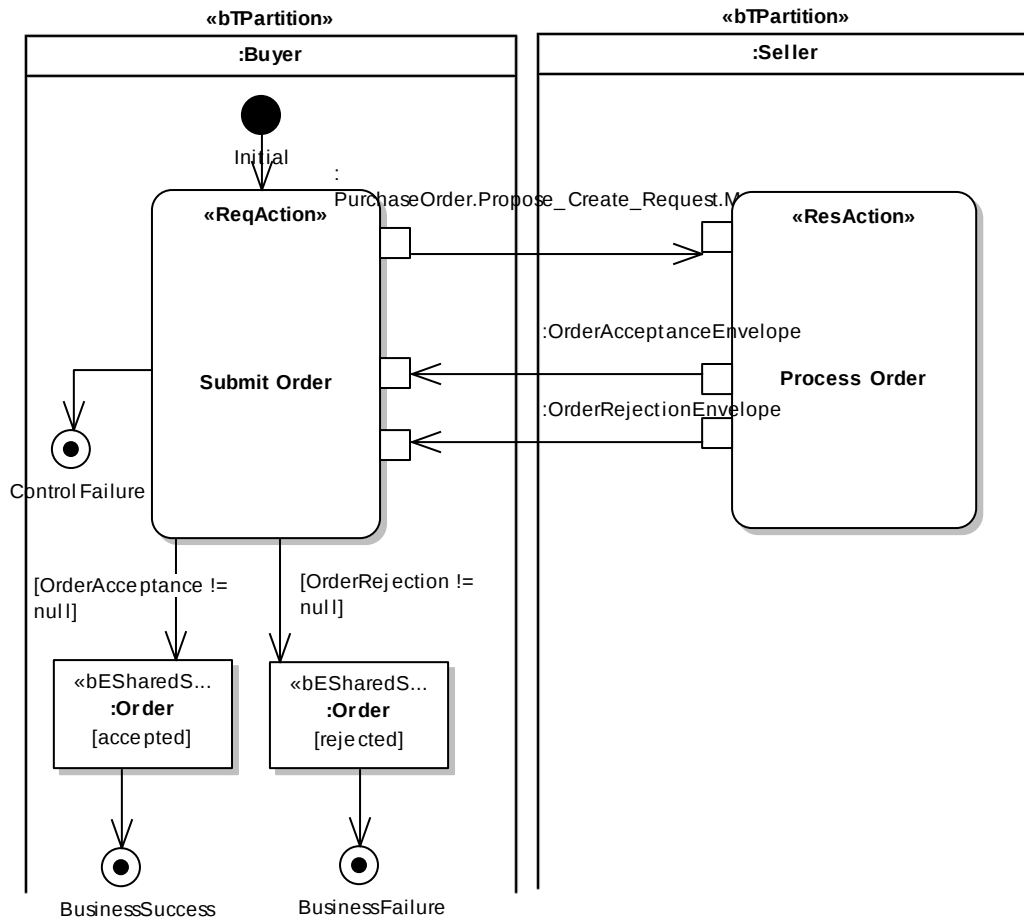


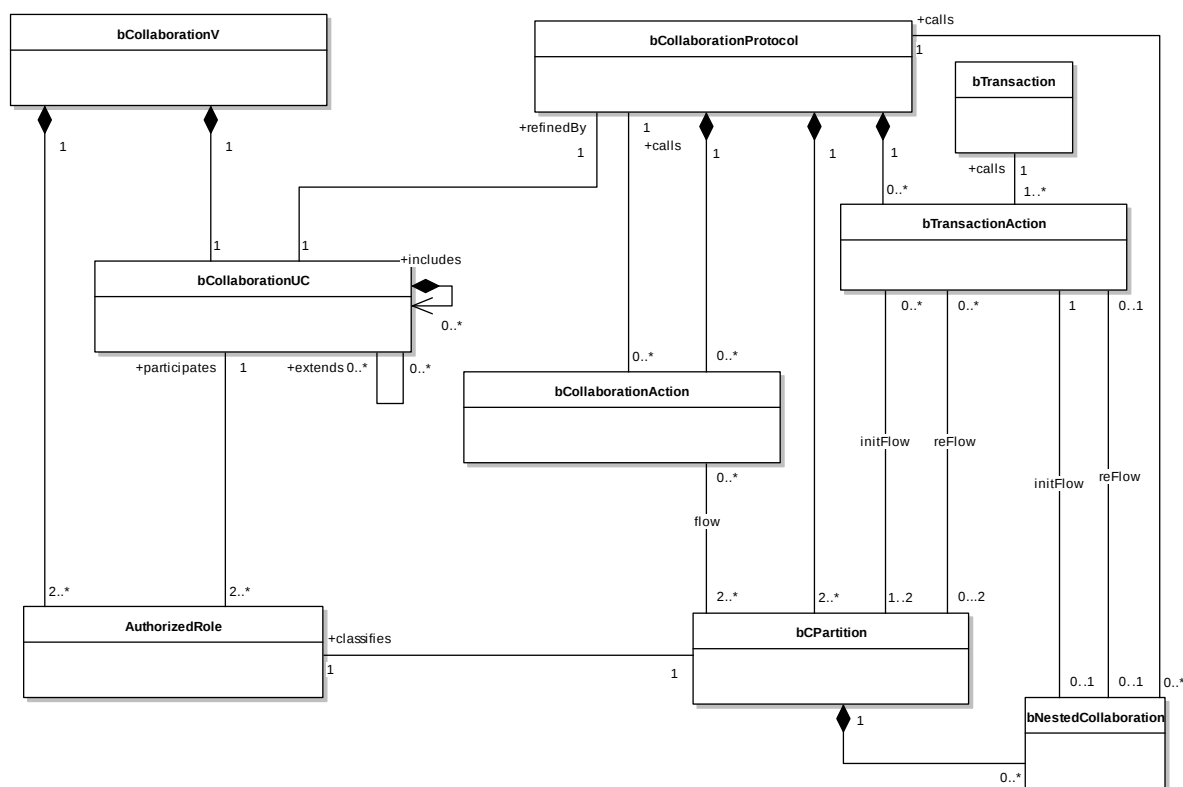
Figure 33 Business Transaction Example: Place Order

5.2.3 Business Collaboration View

5.2.3.1 Abbreviations of Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bCollaborationV	BusinessCollaborationView
AuthorizedRole	AuthorizedRole
bCollaborationUC	BusinessCollaborationUseCase
bCollaborationProtocol	BusinessCollaborationProtocol
bCPartition	BusinessCollaborationPartition
bTransactionAction	BusinessTransactionAction
bCollaborationAction	BusinessCollaborationAction
bNestedCollaboration	NestedBusinessCollaboration

790 5.2.3.2 Conceptual Description (informative)



791

792

Figure 34 Business Collaboration View - Conceptual Overview

793

A *BusinessCollaborationView* is used to define the business choreography of exactly one business collaboration. This business choreography is specified by the concept of a *BusinessCollaborationProtocol*. The requirements of a *BusinessCollaborationProtocol* are captured by a *BusinessCollaborationUseCase*.

795

796

Each *BusinessCollaborationUseCase* and its corresponding *BusinessCollaborationProtocol* are defined in their own business collaboration view package. Accordingly, the *BusinessCollaborationView* is composed of exactly one *BusinessCollaborationUseCase* and one *BusinessCollaborationProtocol*.

798

799

800 5.2.3.2.1 Business Collaboration Use Case Diagram

801

At least two authorized roles participate in a business collaboration use case. These authorized roles must be defined in the same business collaboration view package as the corresponding business collaboration use case. Accordingly, a *BusinessCollaborationView* is composed of two or more *AuthorizedRoles*. This means, if a certain role (e.g. buyer, seller, etc.) participates in multiple business collaborations, it requires a different authorized role for each business collaboration use case. Each authorized role of the same role (i.e., with the same name) is in a different namespace of a corresponding business collaboration view. Therefore, an authorized role participates in only one business collaboration use case – it is the one in the same business collaboration view. Accordingly, *BusinessCollaborationUseCase* and *AuthorizedRole* are related by a 1 to (2..n) association. This association is defined as a *participates* association. It is important to note, that the same authorized role is not associated twice to the same business collaboration use case.

810

811 A business collaboration use case may include additional business collaboration use cases. A business
812 collaboration use case may optionally have multiple parent business collaboration use cases. Hence,
813 *BusinessCollaborationUseCase* has a unary (0..n) to (0..n) include-composition. A business collaboration use
814 case may include multiple business transaction use cases. A business transaction use case must be included
815 in at least one business collaboration use case. Consequently, an (1..n) to (0..n) aggregation between
816 *BusinessCollaborationUseCase* and *BusinessTransactionUseCase* exists. It is important that a business
817 collaboration use case includes at minimum one use case – no matter whether this is a business
818 collaboration use case or a business transaction use case. A hierarchy of business collaboration use cases
819 built by include-compositions must not include any cycles. A business transaction uses case cannot be
820 further decomposed by an include-association.

821 A business collaboration use case may be extended by additional business collaboration use cases. A
822 business collaboration use case may optionally extend multiple business collaboration use cases. Hence,
823 *BusinessCollaborationUseCase* has a unary (0..n) to (0..n) extend-association.

824 5.2.3.2.2 Business Collaboration Protocol Diagram

825 A business choreography view is used to define the business choreography of exactly one business
826 collaboration. The *BusinessCollaborationProtocol* follows exactly the requirements defined by the
827 corresponding *BusinessCollaborationUseCase*. The business collaboration protocol that describes the
828 business collaboration use case is defined as a child beneath. Accordingly, each
829 *BusinessCollaborationUseCase* has exactly one *BusinessCollaborationProtocol* beneath. A business
830 collaboration protocol is a “composite” UML *Activity*. The diagram of a business collaboration protocol is
831 described by a flow of UML *Actions*.

832 A business collaboration protocol defines the collaborative business process between two or more
833 authorized roles. The business collaboration protocol must have a *BusinessCollaborationPartiton* for each of
834 the authorized roles defined in the *BusinessCollaborationView*. Hence, a *BusinessCollaborationProtocol* is
835 composed of two or more *BusinessCollaborationPartitions*. Each *BusinessCollaborationPartition* relates to
836 one and only one *AuthorizedRole* defined in the *BusinessCollaborationView*. Each *AuthorizedRole* in the
837 *BusinessCollaborationView* is assigned to exactly one *BusinessCollaborationPartition*. It follows, that the
838 every *BusinessCollaborationPartition* of a *BusinessCollaborationProtocol* must be assigned to different
839 authorized roles.

840 The collaborative actions of a business collaboration protocol are business collaboration actions and/or
841 business transaction actions. Hence, a *BusinessCollaborationProtocol* is composed of zero to many
842 *BusinessCollaborationActions* and of zero to many *BusinessTransactionActions*. However, at least one
843 business collaboration action or at least one business transaction action must be present in a business
844 collaboration protocol. Transitions defining the flow among the business collaboration activities and/or
845 business transaction activities may be guarded by the states of business entities.

846 A business collaboration action is characterized by the fact that it is executed by calling a business
847 collaboration protocol. This calling behavior is accomplished by classifying the behavior of the business
848 collaboration action by the desired business collaboration protocol. Not every business collaboration
849 protocol is a called by a business collaboration action. A business collaboration protocol may be called by
850 multiple business collaboration actions. Thus, the behavioral classifying relationship between
851 *BusinessCollabortionAction* and *BusinessCollaborationProtocol* is (0..n) to 1.

852 A business transaction action is characterized by the fact that it is executed by calling a business transaction.
853 This calling behavior is accomplished by classifying the behavior of the business transaction action by the
854 desired business transaction. Since the business transaction is a concept of the business transaction view it is
855 described in more detail above. Each business transaction must be at least once used to refine a business
856 transaction action. A business transaction may be called by many business transaction actions. Hence, the
857 behavioral classifying relationship between *BusinessTransactionAction* and *BusinessTransaction* is (1..n) to 1.

858 In many scenarios, there is a requirement for a nested business collaboration within the scope of execution
859 of a given business transaction action. In other words, before a responding authorized role can send a
860 response as required by the business transaction action that calls a two-way business transaction, the
861 responding authorized role has to first participate in a business collaboration with other business partners.
862 UMM supports this scenario by introducing the concept of a *NestedBusinessCollaboration*. Like the business
863 collaboration action, the *NestedBusinessCollaboration* is characterized by the fact that it is executed by
864 calling another business collaboration protocol. This calling behavior is accomplished by classifying the
865 behavior of the *NestedBusinessCollaboration* by the desired business collaboration protocol. Not every
866 business collaboration protocol is called by a *NestedBusinessCollaboration*. A business collaboration
867 protocol may be called by multiple *NestedBusinessCollaborations*. Thus, the behavioral classifying
868 relationship between *NestedBusinessCollaboration* and *BusinessCollaborationProtocol* is (0..n) to 1.
869 However, unlike the business collaboration action, the *NestedBusinessCollaboration* must reside within a
870 business collaboration partition representing the responding authorized role of a given business transaction
871 action. Accordingly, not every business collaboration partition representing a business transaction action
872 responding role includes a *NestedBusinessCollaboration*. Thus, there is a 1 to (0..n) composition between a
873 *BusinessCollaborationPartiton* and a *NestedBusinessCollaboration*.

874 In UMM 2.0, role mapping between business collaboration authorized roles and either called business
875 transaction authorized roles or business collaboration protocol authorized roles is defined in the business
876 collaboration protocol and no longer by the business collaboration use case. This role mapping is
877 accomplished by information flows and specializations of information flows, i.e. *InitiatingFlow* and
878 *RespondingFlow*, between either business collaboration partitions or nested collaborations and either
879 business collaboration actions or business transaction actions. Using the approach also enhances the
880 business collaboration protocol by graphically illustrating the relationships between authorized roles and the
881 choreography of actions within a business collaboration protocol. This mapping approach is defined by the
882 following cases:

883 1. *From business collaboration partitions to business collaboration actions*. The UML 2.0 Information
884 flow is used. The source of Information flow must be the business collaboration partition and the
885 target must be the business collaboration action. This means that the authorized role in this business
886 collaboration protocol is participating in the called business collaboration protocol. Therefore the
887 calling business collaboration protocol must have at least the same number of business collaboration
888 partitions as the number of business collaboration partitions in the called business collaboration
889 protocol. There are two cases for defining a role mapping:

- 890 • The authorized role name of the calling business collaboration protocol maps to an
891 authorized role in the called business collaboration protocol which has exactly the same
892 name. In this case, the information flow association explicitly defines the role mapping
893 between these two roles.

- 894 • The authorized role name of the calling business collaboration protocol maps to an
895 authorized role in the called business collaboration protocol which has a different name. In
896 this case, the information flow association must be classified by the authorized role of the
897 called business collaboration protocol.
- 898 2. *From business collaboration partitions to business transaction actions.* In this case two
899 specializations of the Information flow are used, *InitiatingFlow* and *RespondingFlow*. In all cases an
900 authorized role of a business collaboration protocol initiates a business transaction action, which
901 calls a business transaction, which also has an initiating authorized role. To provide this role mapping
902 the business collaboration partition classified by an authorized role is the source of the
903 *InitiatingFlow* and the business transaction action is the target. Likewise for two-way business
904 transactions, a business collaboration partition is the source of the *RespondingFlow* and the business
905 transaction action is the target.
- 906 3. *From business transaction actions to business collaboration partitions.* In this case two
907 specializations of the Information Flow are used, *InitiatingFlow* and *RespondingFlow*. In all cases an
908 authorized role of a business collaboration protocol responds to a business transaction action which
909 calls a business transaction which also has a responding authorized role. To provide this role
910 mapping the business transaction action is the source of an *InitiatingFlow* and a business
911 collaboration partition is the target. Therefore the authorized role that classifies the business
912 collaboration partition maps to the responding business transaction authorized role. Likewise, for
913 two-way business transactions, a business transaction action is the source of a *RespondingFlow* and
914 a business collaboration partition is the target.
- 915 4. *From business transaction actions to nested business collaborations.* In this case only one
916 specialization of the Information Flow is used, *InitiatingFlow*. The source of the *InitiatingFlow* is a
917 business transaction action and the target is a nested business collaboration. This means that the
918 responding authorized role of the business collaboration initiates the business collaboration protocol
919 called by the nested business collaboration and therefore maps to initiating authorized role of the
920 called business collaboration protocol.
- 921 5. *From nested business collaborations to business transaction actions.* In this case only one
922 specialization of the Information Flow is used, *RespondingFlow* and only applies in the case of two-
923 way business transactions. The *RespondingFlow* indicates that the called business collaboration has
924 completed and the responding authorized role can now return the response envelope to the
925 initiating authorized role. The *NestedBusinessCollaboration* is the source of the *RespondingFlow* and
926 a business transaction action is the target.

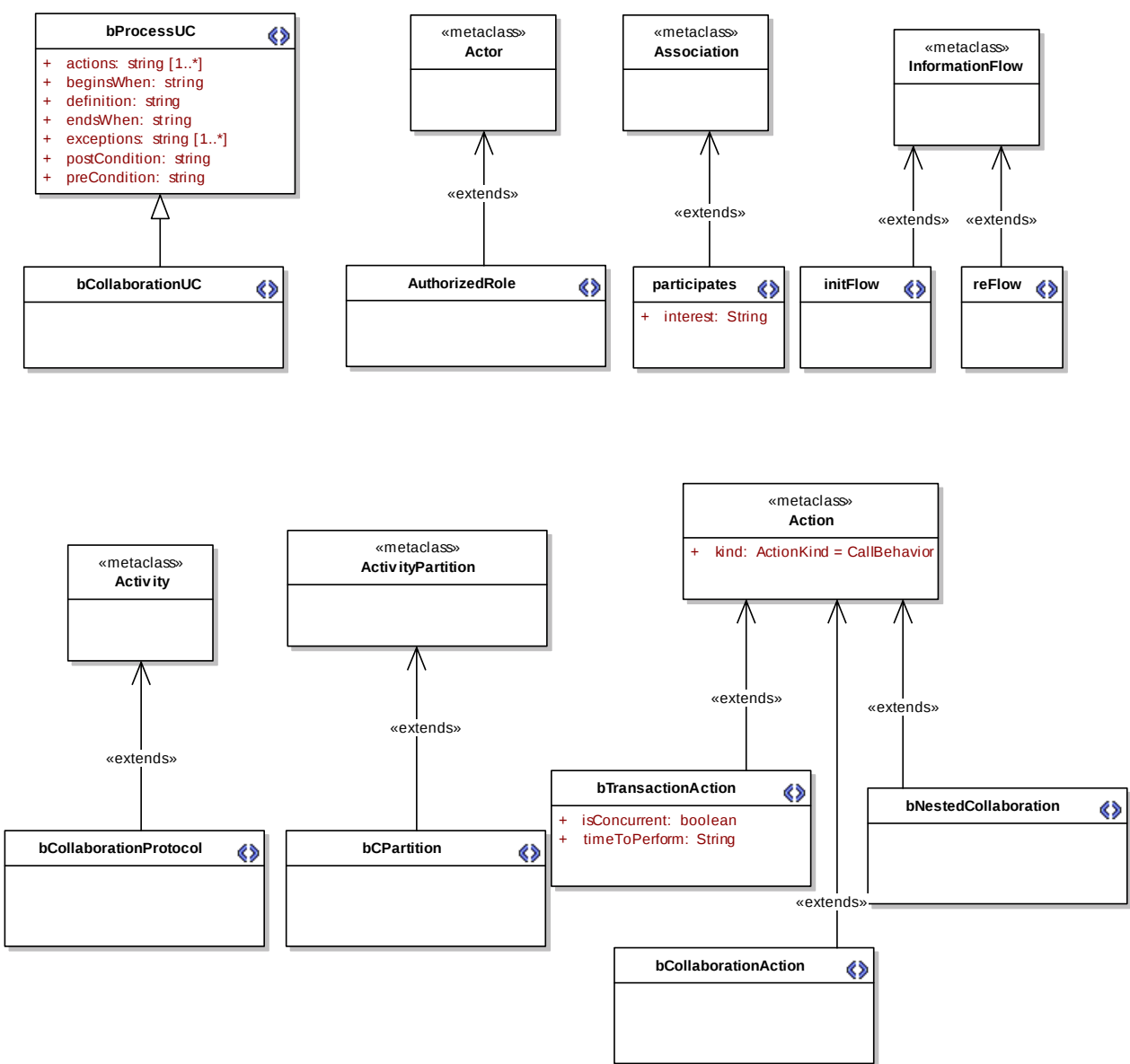


Figure 35 Business Collaboration View - Abstract Syntax

Stereotype	bCollaborationUC (BusinessCollaborationUseCase)
Base Class	UseCase
Parent	bProcessUC (Business Process Use Case)
Description	A business collaboration use case describes in detail the requirements on the collaboration between two or more involved partners. Business partner types take part in a business collaboration use case by playing an authorized role in it. A business collaboration use case can be broken down into further business collaboration use cases and business transaction use cases. A business collaboration use case may extend another business collaboration use case.
Tag Definition	Inherited tagged values:

931

	- definition - beginsWhen - preCondition - endsWhen - postCondition - exceptions - actions
--	--

932

Stereotype	AuthorizedRole
Base Class	Actor
Parent	N/A
Description	Already defined before in previous sub-section
Tag Definition	No tagged values.

933

Stereotype	bCollaborationProtocol (BusinessCollaborationProtocol)
Base Class	Activity
Parent	N/A
Description	A business collaboration protocol choreographs business transaction actions and/or business collaboration actions. At least one action of either one must be present. A business collaboration protocol is a long running transaction that does not meet the atomic principle of transactions. It should be used in cases where transaction rollback is inappropriate.
Tag Definition	No Tagged Values

934

Stereotype	bCPartition (BusinessCollaborationPartition)
Base Class	ActivityPartition
Parent	N/A
Description	A business collaboration partition is used to define an area of responsibility. The business collaboration partition is always classified by an authorized role defined as a participant in the corresponding business collaboration use case. A business collaboration partition may be empty. It is not empty in the special case of a nested collaboration. A nested collaboration must be placed within the business collaboration partition of the authorized role which is the responding authorized role in the triggering business transaction action and which will initiate the nested collaboration.
Tag Definition	No tagged values.

935

Stereotype	bTransactionAction (BusinessTransactionAction)
------------	--

Base Class	Action with call behaviour action kind (CallBehaviorAction)	
Parent	N/A	
Description	A business transaction action is an action within a business collaboration protocol. This action is refined by a using the call behaviour to classify the behaviour of this business transaction action by one and only one business transaction. The business transaction action executes the called business transaction. The business transaction action can be executed more than once at the same time if the “isConcurrent” property is true.	
Tag Definition	timeToPerform	
	Type	TimeExpression
	Multiplicity	1
	Description	A business transaction action has to be executed within a specific duration. The initiating partner must send a failure notification to a responding partner on timeout. A responding partner simple terminates its activity. The time to perform is the maximum duration between the moment the requesting authorized role initiates the business transaction action, i.e. sending the requesting business information envelope, and the moment the requesting authorized role receives a substantive response. The substantive response is the responding business information envelope if there is any. In case not, it is the acknowledgement of processing, if any. If not it is the acknowledgement of receipt, if any.
	isConcurrent	
	Type	Boolean
	Multiplicity	1
	Description	If the business transaction action is concurrent then more than one business transaction action of the same underlying business transaction can be open at one time in executing the same business collaboration with the same business partner type. If the business transaction action is not concurrent then only one business transaction action of the same underlying business transaction can be open at one time.

936

Stereotype	bCollaborationAction (BusinessCollaborationAction)	
Base Class	Action with call behaviour action kind (CallBehaviorAction)	
Parent	N/A	
Description	A business collaboration action is an action within a business collaboration protocol. This business collaboration action is refined by using the call behaviour to classify the behaviour of this business collaboration action by one and only one business collaboration protocol. The business choreography action executes the called business collaboration protocol exactly once. It follows, that business collaboration protocols might be recursively nested.	
Tag Definition	No Tagged Values	

937

Stereotype	NestedBCollaboration (NestedBusinessCollaboration)
-------------------	---

938

Base Class	Action with call behaviour action kind (CallBehaviorAction)
Parent	N/A
Description	A nested business collaboration represents the case where the responding authorized role of a business transaction action after receiving a requesting information envelope which is represented as an initFlow (InitiatingFlow) must carry out an additional business collaboration protocol with other business partners before responding to the initiating authorized role of a given business transaction action indicated by a reFlow (RespondingFlow). This nested business collaboration is refined by using the call behaviour to classify the behaviour of this nested business collaboration by one and only one business collaboration protocol. The nested collaboration executes the called business collaboration protocol exactly once.
Tag Definition	No tagged values.

939

Stereotype	initFlow (InitiatingFlow)
Base Class	Information Flow
Parent	N/A
Description	The initiating flow represents the following two cases: • The initiating flow of information that triggers the execution of a business transaction action. The source of the initiating flow is the business collaboration partition that is classified by the authorized role initiating the business transaction action and the target is the business transaction action. In this case the initiating flow provides the authorized role mapping between the initiating role of the business transaction action and the initiating authorized role of the business transaction that is called by this business transaction action. • The initiating flow of information that triggers an execution on the responder's side. The source of this initiating flow is the business transaction action and the target is the business collaboration partition which is classified by the authorized role responding in the business transaction action. In this case the initiating flow provides the authorized role mapping between the responding role of the business transaction action and the responding authorized role of the business transaction that is called by this business transaction action. In the special case that the initiating flow triggers a nested business collaboration, the target of the initiating flow is not the business collaboration partition, but the nested business collaboration residing within this business collaboration partition.
Tag Definition	No tagged values.

Stereotype	reflow (RespondingFlow)
Base Class	Information Flow
Parent	N/A
Description	The responding flow in case of two-way transactions represents the following two cases: • The responding flow of information that completes the execution of a business transaction action. The source of the responding flow is the business collaboration partition that is classified by the authorized role responding in the business transaction action and the target is the business transaction action. In the special case that the responding flow is started after a nested business collaboration has completed, the source of the responding flow is not the business

	collaboration partition, but the nested business collaboration residing within this business collaboration partition. The responding flow of information that completes the business transaction action on the initiator's side. The source of this initiating flow is the business transaction action and the target is the business collaboration partition which is classified by the authorized role initiating the business transaction action.
Tag Definition	No tagged values.

5.2.3.4 Constraints (normative)

- C.58. A *BusinessCollaborationView* MUST contain exactly one *BusinessCollaborationUseCase*.
- C.59. A *BusinessCollaborationView* MUST contain two to many *AuthorizedRoles*.
- C.60. A *BusinessCollaborationUseCase* MUST have two to many *participates* associations to *AuthorizedRoles* contained in the same *BusinessCollaborationView*.
- C.61. Each *AuthorizedRole* contained in the *BusinessCollaborationView* MUST have exactly one *participates* association to the *BusinessCollaborationUseCase* included in the same *BusinessCollaborationView*.
- C.62. A *BusinessCollaborationUseCase* MUST have one to many *include* relationships to another *BusinessCollaborationUseCase* or to a *BusinessTransactionUseCase*.
- C.63. Exactly one *BusinessCollaborationProtocol* MUST be placed beneath each *BusinessCollaborationUseCase*.
- C.64. A *BusinessCollaborationProtocol* MUST contain one to many *BusinessTransactionActions* and/or *BusinessCollaborationAction*.
- C.65. Each *BusinessTransactionAction* MUST call exactly one *BusinessTransaction*
- C.66. Each *BusinessTransaction* called by a *BusinessTransactionAction* MUST be placed beneath a *BusinessTransactionUseCase* which is included in the *BusinessCollaborationUseCase* that covers the corresponding *BusinessCollaborationProtocol*.
- C.67. Each *BusinessCollaborationProtocol* called by a *BusinessCollaborationAction* MUST be placed beneath a *BusinessCollaborationProtocolUseCase* which is included in the *BusinessCollaborationUseCase* that covers the corresponding *BusinessCollaborationProtocol*.
- C.68. A *BusinessCollaborationProtocol* MUST contain two to many *BusinessCollaborationPartitions*.
- C.69. The number of *AuthorizedRoles* in the *BusinessCollaborationView* MUST match the number of *BusinessCollaborationPartitions* in the *BusinessCollaborationProtocol* which is placed beneath the *BusinessCollaborationUseCase* of the same *BusinessCollaborationView*.
- C.70. Each *AuthorizedRole* in the *BusinessCollaborationView* MUST be assigned to a *BusinessCollaborationPartition* in the *BusinessCollaborationProtocol* which is placed beneath the *BusinessCollaborationUseCase* of the same *BusinessCollaborationView*.
- C.71. Each *BusinessCollaborationPartition* MUST be classified by exactly one *AuthorizedRole* included in the same *BusinessCollaborationView* as the *BusinessCollaborationUseCase* covering the *BusinessCollaborationProtocol* containing this *BusinessCollaborationPartition*.
- C.72. A *BusinessCollaborationPartition* MUST be either empty or contain one to many *NestedBusinessCollaborations*.
- C.73. Each *BusinessTransactionAction* MUST be the target of exactly one *InitialFlow* which source MUST be a *BusinessCollaborationPartition*.
- C.74. Each *BusinessTransactionAction* MUST be the source of exactly one *InitialFlow* which target MUST be either a *BusinessCollaborationPartition* or a *NestedBusinessCollaboration*.

- 977 C.75. The *InitialFlow* sourcing from a *BusinessTransactionAction* and the *InitialFlow* targeting a
978 *BusinessTransactionAction* MUST NOT be targeting to / sourcing from the same
979 *BusinessCollaborationPartition*, nor targeting to a *NestedBusinessCollaboration* within the same
980 *BusinessCollaborationPartition*.
- 981 C.76. If a *BusinessTransactionAction* calls a two-way *BusinessTransaction*, this
982 *BusinessTransactionAction* MUST be the source of exactly one *RespondingFlow* which target MUST
983 be a *BusinessCollaborationPartition*.
- 984 C.77. If a *BusinessTransactionAction* calls a two-way *BusinessTransaction*, this
985 *BusinessTransactionAction* MUST be the target of exactly one *RespondingFlow* which source MUST
986 be either a *BusinessCollaborationPartition* or a *NestedBusinessCollaboration*.
- 987 C.78. The *RespondingFlow* sourcing from a *BusinessTransactionAction* and the *RespondingFlow*
988 targeting a *BusinessTransactionAction* MUST NOT be targeting to /sourcing from the same
989 *BusinessCollaborationPartition*, nor targeting to a *NestedBusinessCollaboration* within the same
990 *BusinessCollaborationPartition*.
- 991 C.79. If a *BusinessTransactionAction* calls a one-way *BusinessTransaction*, this
992 *BusinessTransactionAction* MUST NOT be the source of a *RespondingFlow* and MUST NOT be the
993 target of a *RespondingFlow*.
- 994 C.80. The *RespondingFlow* targeting a *BusinessTransactionAction* must start from the
995 *BusinessCollaborationPartition* / *NestedBusinessCollaboration* which is the target of the *InitialFlow*
996 starting from the same *BusinessTransactionAction*.
- 997 C.81. The *RespondingFlow* starting from a *BusinessTransactionAction* must target the
998 *BusinessCollaborationPartition* which is the source of the *InitialFlow* targeting to the same
999 *BusinessTransactionAction*.
- 1000 C.82. A *NestedBusinessCollaboration* MUST be the target of exactly one *InitialFlow*.
- 1001 C.83. A *NestedBusinessCollaboration* MAY be the source of a *RespondingFlow*, but MUST NOT be
1002 the source of more than one *RespondingFlow*.
- 1003 C.84. A *BusinessCollaborationAction* MUST be the target of two to many *InformationFlows* (UML
1004 standard: <<flow>>).
- 1005 C.85. A *BusinessCollaborationAction* MUST not be the source of an *InformationFlow*.
- 1006 C.86. A *BusinessCollaborationAction* MUST not be the source and MUST not be the target of an
1007 *InitialFlow*.
- 1008 C.87. A *BusinessCollaborationAction* MUST not be the source and MUST not be the target of a
1009 *RespondingFlow*.
- 1010 C.88. A *BusinessTransactionAction* MUST not be the source and MUST not be the target of an
1011 *InformationFlow* (<<flow>>) that is neither stereotyped as *InitialFlow* nor as *RespondingFlow* nor is
1012 of type <<flow>>.
- 1013 C.89. A *NestedBusinessCollaboration* MUST not be the source and MUST not be the target of an
1014 *InformationFlow* that targets to / sources from a *BusinessCollaborationAction*.
- 1015 C.90. The number of *InformationFlows* targeting a *BusinessCollaborationAction* MUST match the
1016 number of *BusinessCollaborationPartitions* contained in the *BusinessCollaborationProtocol* that is
1017 called by this *BusinessCollaborationAction*.
- 1018 C.91. Either an *AuthorizedRole* classifying a *BusinessCollaborationPartition* that is the source of an
1019 *InformationFlow* (UML standard: <<flow>>) targeting a *BusinessCollaborationAction* MUST match an
1020 *AuthorizedRole* classifying a *BusinessCollaborationPartition* in the *BusinessCollaborationProtocol*
1021 that is called by this *BusinessCollaborationAction* or the *InformationFlow* must be classified by an

1022 AuthorizedRole classifying a *BusinessCollaborationPartition* in the *BusinessCollaborationProtocol*
 1023 that is called by this *BusinessCollaborationAction*.

1024 **5.2.3.5 Worksheets**

Form for Business Collaboration Use Case	
General	
Name	
Description	
Business Library Information	
URI	
BusinessTerm	
Version	
Status	
Owner	
Copyright	
Reference(s)	
Participants	
Participating Role	
Participating Role	
[add more participating roles in case of a multiparty collaboration]	
Is Included In (Name of parent Business Collaboration – if there is any)	
Start /End Characteristics	
Affected Business Entities	
Pre-condition	
Post-condition	
Begins When	

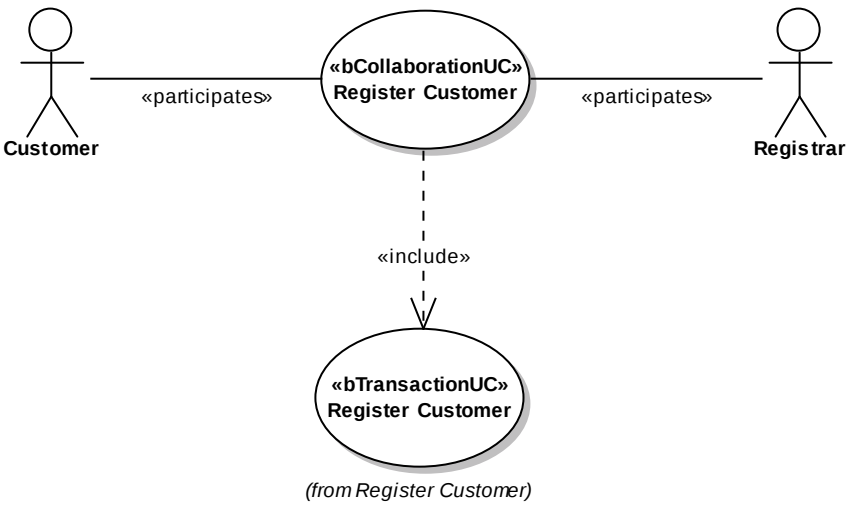
Ends When	
Exceptions	
Included Business Transaction Use Cases (add more Business Transaction Use Cases if needed)	
Business Transaction Use Case Name	
Business Transaction Use Case Name	
Business Transaction Use Case Name	
Business Transaction Use Case Name	

1025

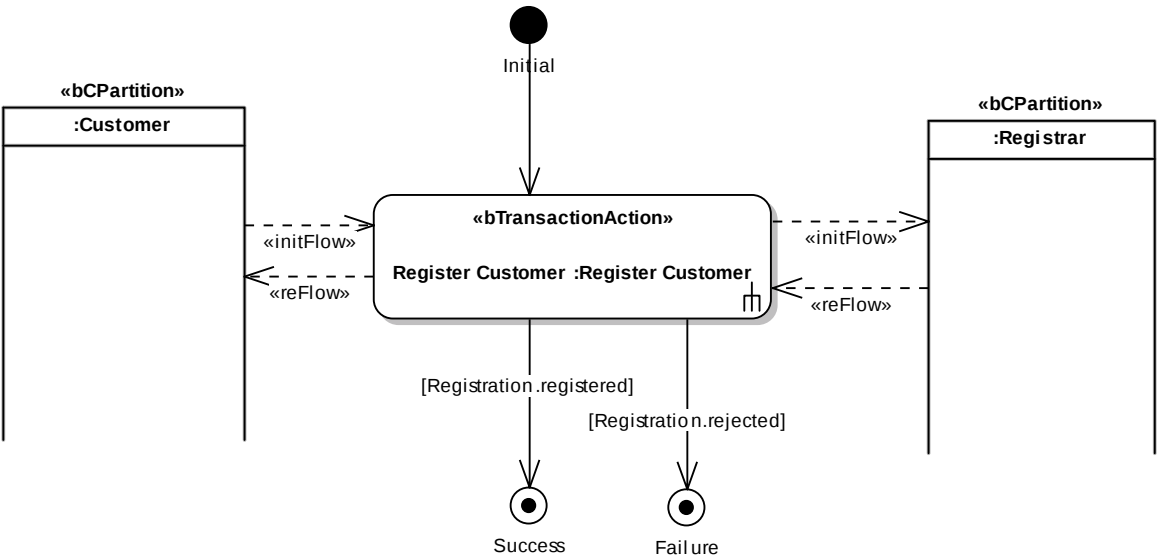
Form for Business Collaboration Protocol	
General	
Name	
Description	
Participants (copy from Business Collaboration Use Case Worksheet)	
Participating Role	
Participating Role	
[add more participating roles if elicited in the Business Collaboration Use Case Worksheet]	
Included Business Transaction Actions / Business Collaboration Actions	
Business Transaction Action	
Name	
Preceding Action(s) including transition condition	
Initiating Role	[select one participating role from above]
Reacting Role	[select one participating role from above]

Business Transaction Action			
Name			
Preceding Action(s) including transition condition			
Initiating Role	[select one participating role from above]		
Reacting Role	[select one participating role from above]		
Business Transaction Action [add more if needed]			
Name			
Preceding Action(s) including transition condition			
Initiating Role	[select one participating role from above]		
Reacting Role	[select one participating role from above]		
Business Collaboration Action [delete if not required or add more if needed]			
Name			
Preceding Action(s) including transition condition			
Role Mapping	Role in this Business Collaboration	Role in the nested Business Collaboration	
Role Mapping	Role in this Business Collaboration	Role in the nested Business Collaboration	
[add more Role Mappings if required]			

1026



1028
1029 **Figure 36 Business Collaboration Use Case Example: Register Customer**



1030
1031 **Figure 37 Business Collaboration Protocol Example: Register Customer**

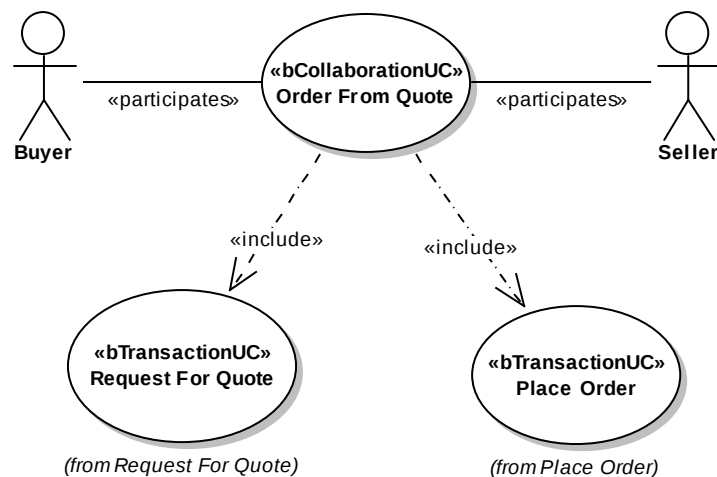


Figure 38 Business Collaboration Use Case Example: Order From Quote

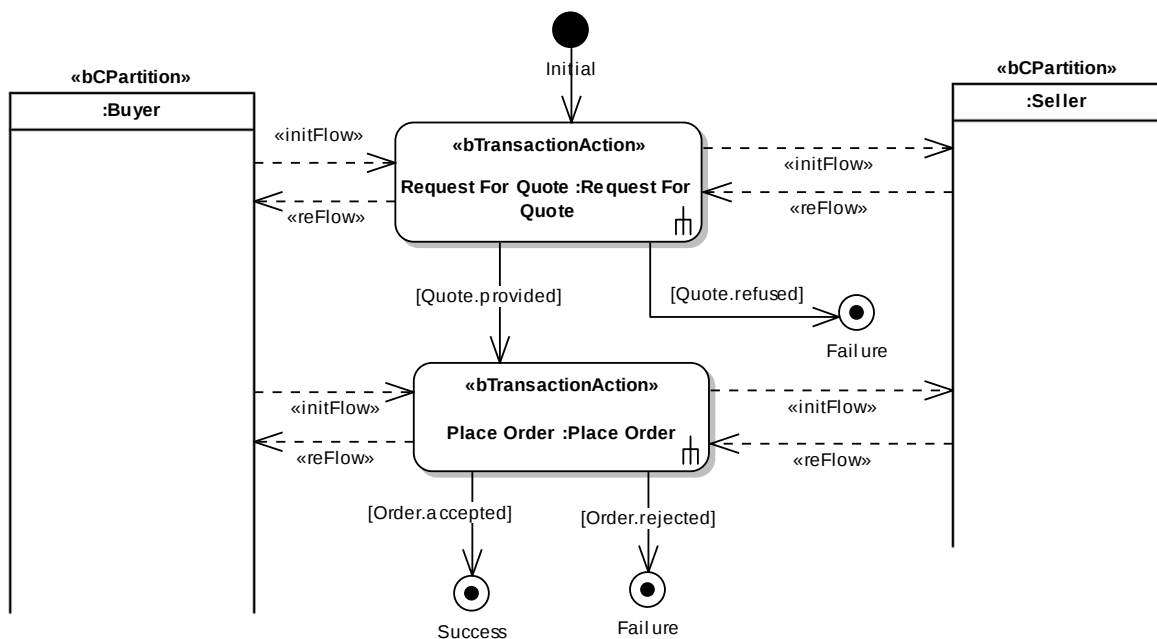


Figure 39 Business Collaboration Protocol Example: Request From Quote

5.2.3.7 Nested Business Collaboration View Example (informative)

In this example the Order from Quote example is modified to show an example of a Nested Collaboration. Before the Seller can either accept or reject the Buyer's order, the seller must confirm the order with his/her business partners. If this confirmation is unsuccessful, it follows that the Seller will respond by sending a negative response message (e.g. *OrderRejectEnvelope*) to the Buyer.

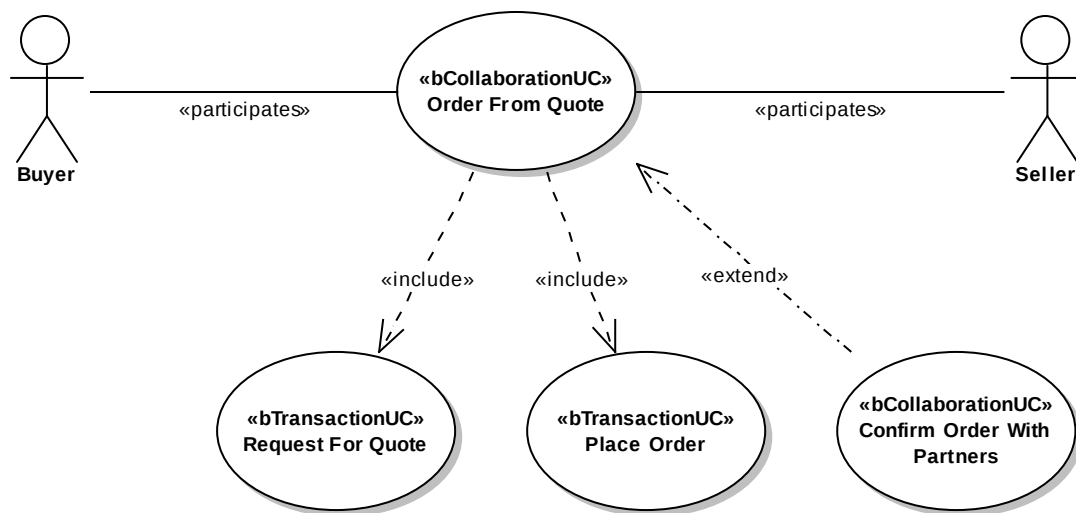


Figure 40 Business Collaboration Use Case Example: Order From Quote

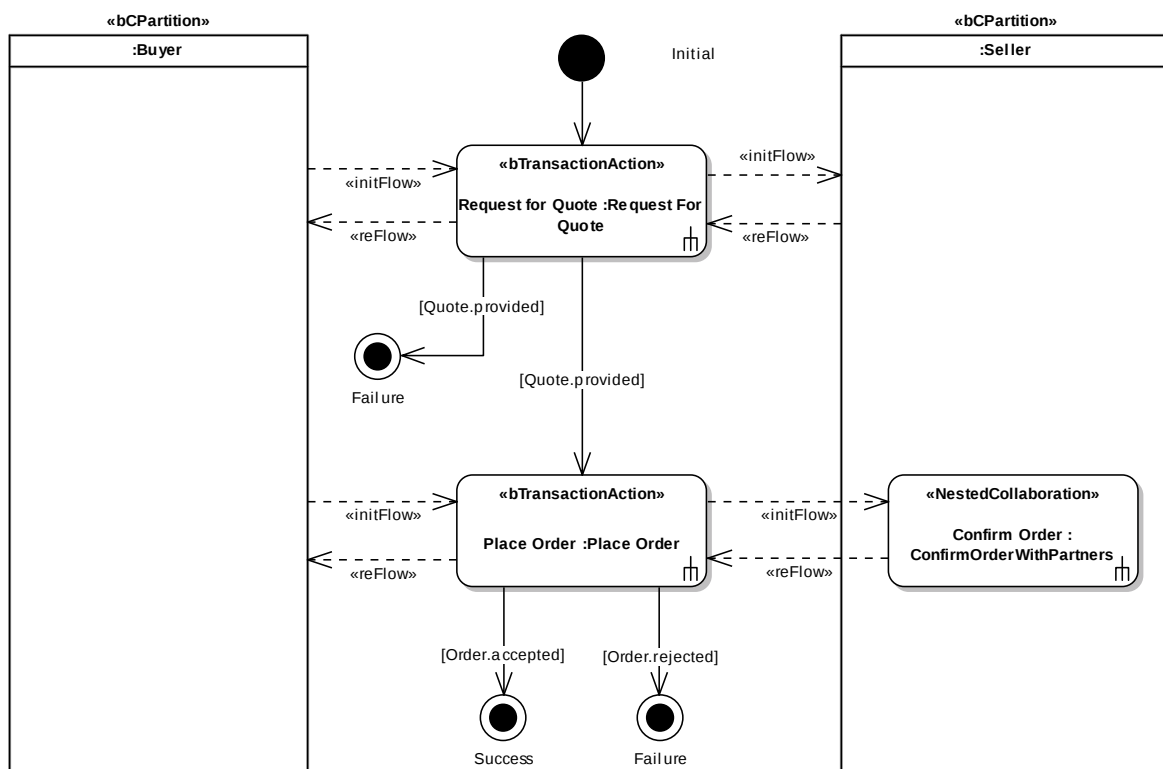


Figure 41 Business Collaboration Protocol Example: Request For Quote

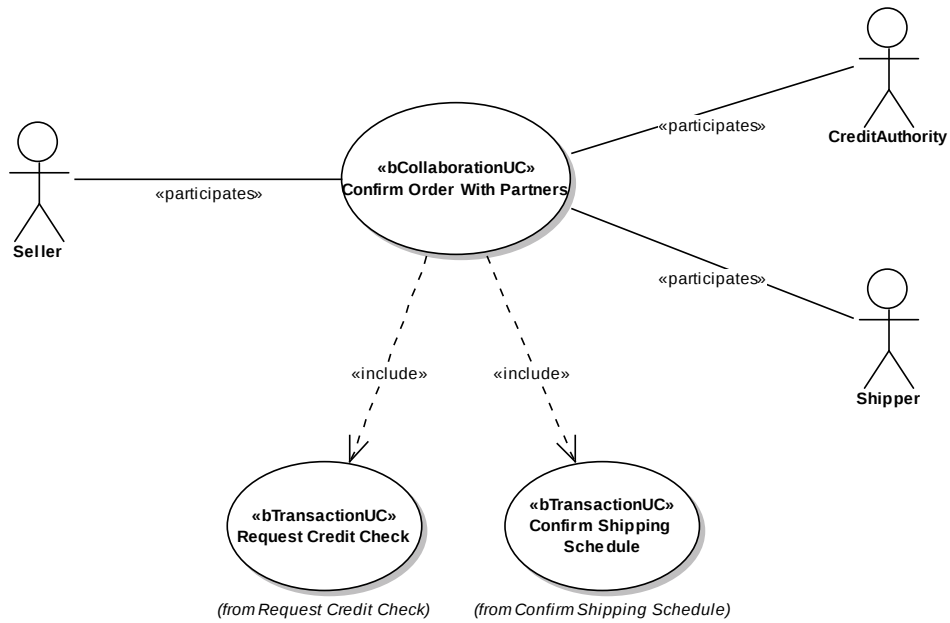


Figure 42 Business Collaboration Use Case Example: Confirm Order With Partners

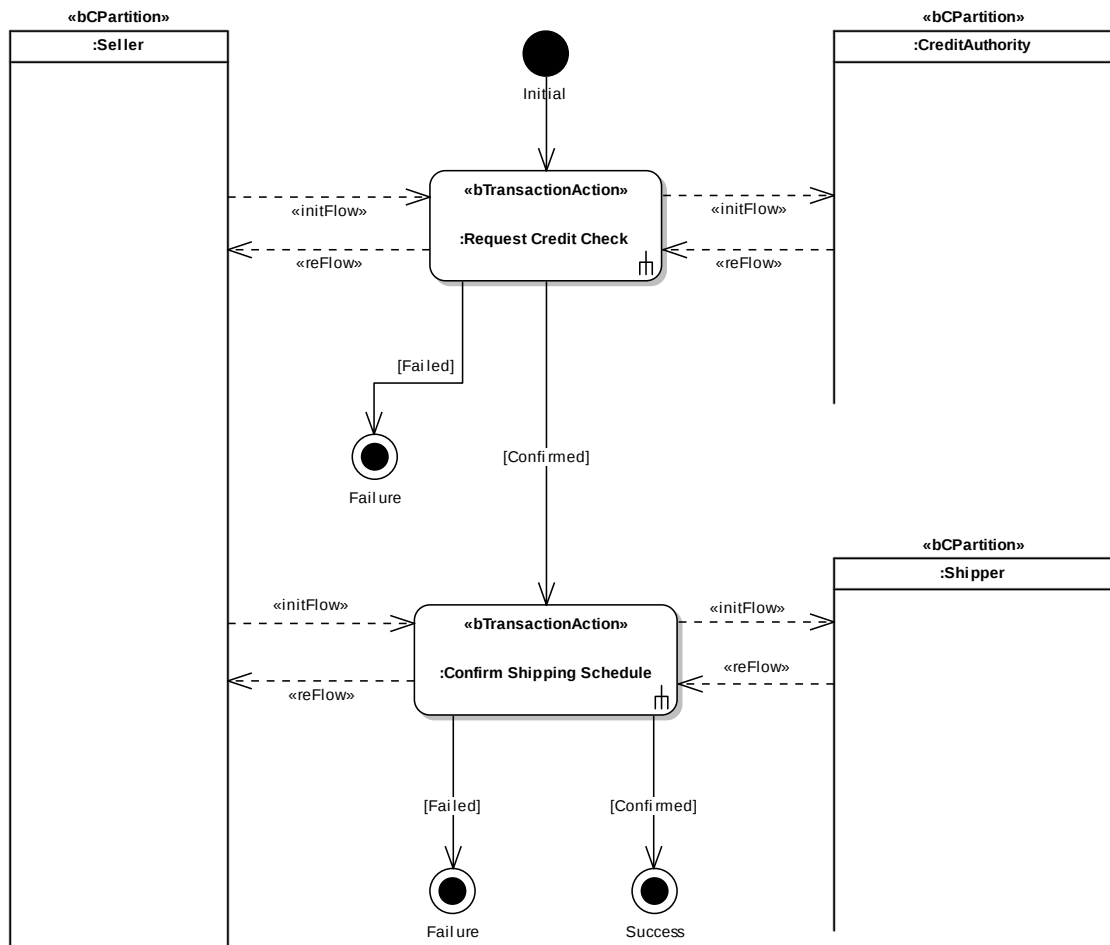


Figure 43 Business Collaboration Protocol Example: Confirm Order with Partners

5.2.4 Business Realization View

5.2.4.1 Abbreviations and Stereotypes

Stereotype Abbreviation	Full Stereotype Name
bChoreographyV	BusinessChoreographyView
bRealizationV	BusinessRealizationView
bRealizationUC	BusinessRealizationUseCase
bCollaborationUC	BusinessCollaborationUseCase
bPartner	BusinessPartner
bProcessUC	BusinessProcessUseCase

5.2.4.2 Conceptual Description (informative)

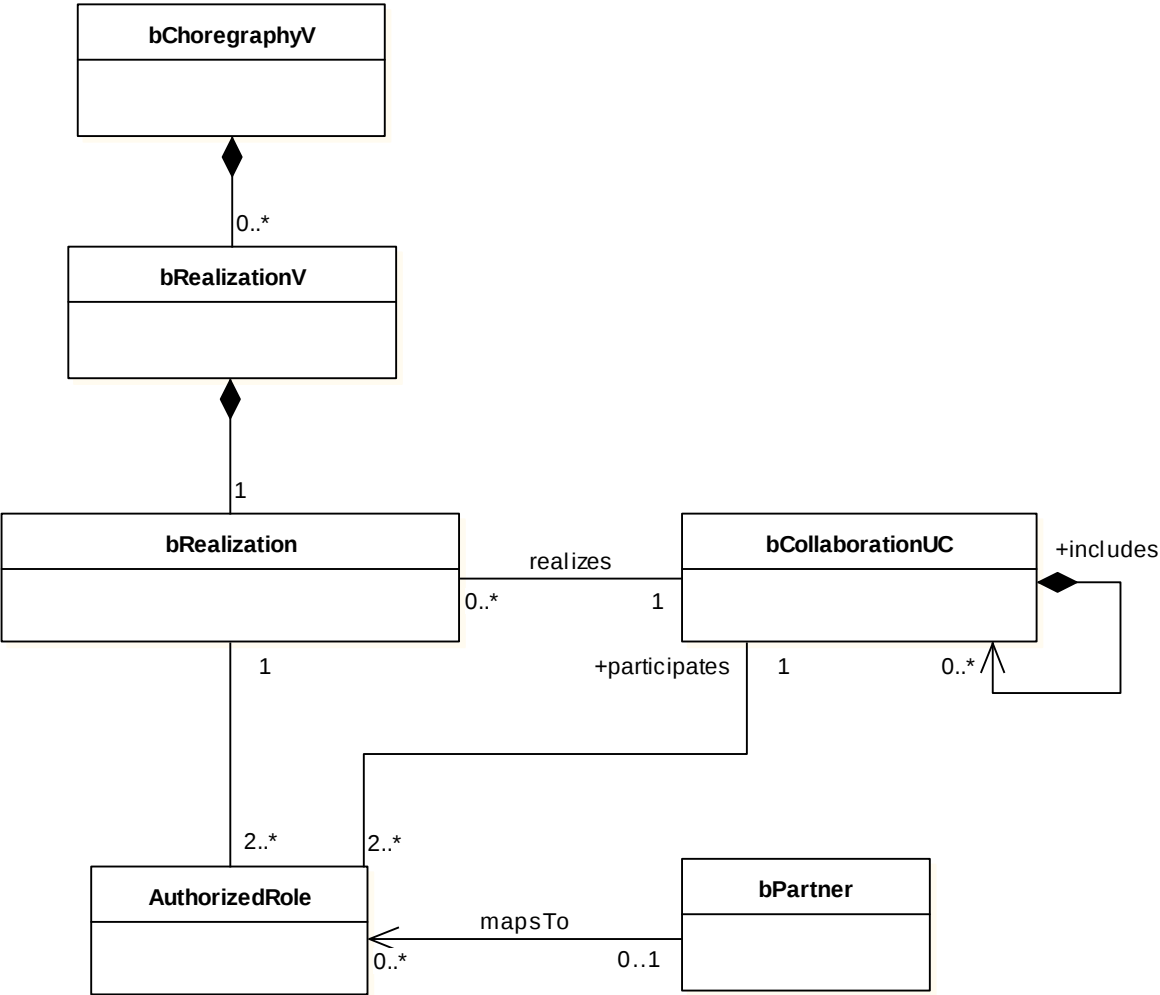


Figure 44 BusinessRealizationView - Conceptual Overview

Business partners identified in the previous business requirements view must not directly be associated with business collaboration use cases and business transaction use cases.

1063 In order to specify that a specific set of business partners collaborate, we use the concept of a business
1064 realization. Each business realization is defined in its own business realization view. Accordingly, the
1065 *BusinessRealizationView* is composed of exactly one *BusinessRealization*. A business realization realizes
1066 exactly one business collaboration use case. Each business collaboration use case may be realized by
1067 multiple business realizations. Not each business collaboration use case (e.g. one that is nested within
1068 another one) needs to have a corresponding business realization. As a consequence, the *realizes*-association
1069 between a *BusinessCollaborationUseCase* and *BusinessRealization* is a 1 to (0..n).

1070 Two or more authorized roles participate in a business realization. These authorized roles (e.g. seller, payee)
1071 must be defined in the same business realization view package as the corresponding business realization.
1072 Accordingly a *BusinessRealizationView* is composed of two or more *AuthorizedRoles*. Usually, the names of
1073 the authorized roles participating in the business collaboration use case (e.g. payer and payee) will be the
1074 names of the authorized roles in the business realization (e.g. payer and payee) realizing it. However, the
1075 authorized roles participating in the business collaboration use case and the business realization will be
1076 defined in different namespaces – each in the package of the corresponding view. In Figure 46 the
1077 authorized role *Buyer* on the lower left hand side participates in the business collaboration use case. It is
1078 defined in a different namespace than the *Buyer* participating in the business realization.

1079 Similar to the business collaboration use case, the *BusinessRealization* and *AuthorizedRole* are related by an
1080 1 to (2..n) association. Furthermore, the number of actors participating in a business collaboration use case
1081 must be the same as the number of actors participating in the business realization realizing it.

1082 In order to bind a business realization to the business partners executing it, business partners are mapped to
1083 the authorized roles participating in the business realization. It is required that each authorized role of a
1084 business realization (but not an authorized role in general) is target of exactly one *mapsTo*-association from
1085 a business partner. A business partner may play multiple authorized roles of a business realization.
1086 Consequently, there is a (0..1) to (0..n) *mapsTo*-association between *BusinessPartner* and *AuthorizedRole*.

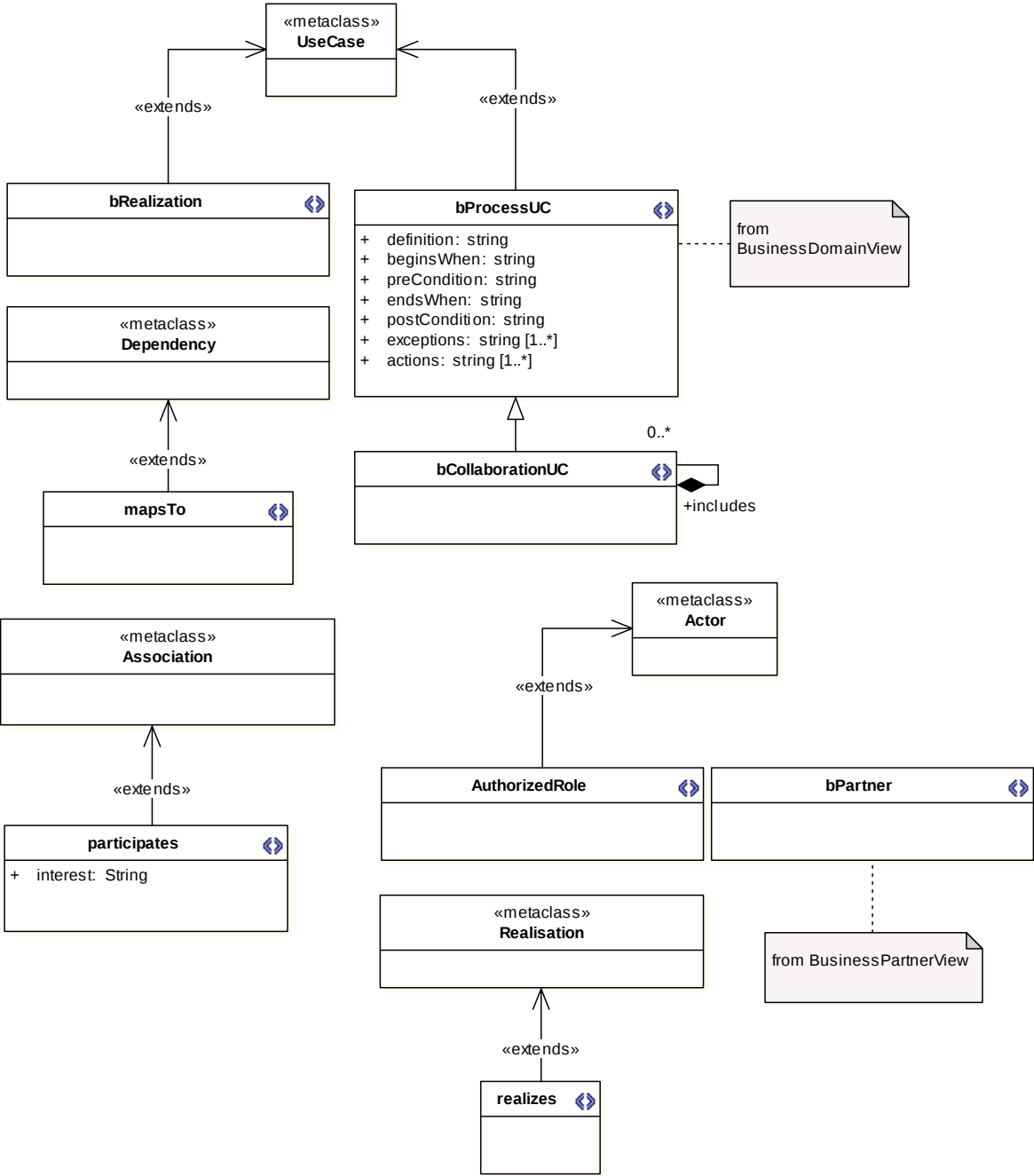


Figure 45 BusinessRealizationView - Abstract Syntax

Stereotype	bRealization (BusinessRealization)
Base Class	UseCase
Parent	N/A
Description	A business realization realizes a business collaboration use case between a specific set of business partners. The requirements of the business realization are the ones defined in the tags of the corresponding business collaboration use case. Thus, the business realization does not include any tag

	definitions for capturing requirements.
Tag Definition	No tagged values

1091

Stereotype	AuthorizedRole
Base Class	Actor
Parent	N/A
Description	An authorized role (e.g. a “buyer”) is a concept which is more generic than a business partner (e.g. a “broker”) and allows the reuse of collaborations by mapping an <i>AuthorizedRole</i> to a business partner within a given scenario. Since business collaboration use case and business transaction use case are defined as occurring between authorized roles, they might be reused by different business partners (a “broker” or a “custodian”) in different scenarios of the same domain or even in different domains.
Tag Definition	No tagged values.

1092

Stereotype	mapsTo
Base Class	Dependency
Parent	N/A
Description	A mapsTo dependency represents (1) the fact, that a business partner plays a certain authorized role in a business realization and (2) the fact, that an authorized role of a source business collaboration use case takes on a certain authorized role in a target business transaction use case or business collaboration use case.
Tag Definition	No tagged values.

1093

1094 5.2.4.4 Constraints (normative)

1095 C.92. A *BusinessRealizationView* MUST contain exactly one *BusinessRealization*, two to many
1096 *AuthorizedRoles*, and two to many *participates* associations.

1097 C.93. A *BusinessRealization* MUST be associated with two to many *AuthorizedRoles* via
1098 stereotyped binary *participates* associations.

1099 C.94. A *BusinessRealization* MUST be the source of exactly one realization dependency to a
1100 *BusinessCollaborationUseCase*.

1101 C.95. A *BusinessRealization* MUST NOT be the source or target of an *include* or *extends*
1102 association.

1103 C.96. All dependencies from/to an *AuthorizedRole* must be stereotyped as *mapsTo*.

1104 C.97. An *AuthorizedRole*, which participates in a *BusinessRealization*, must be the target of exactly
1105 one *mapsTo* dependency starting from a *BusinessPartner*. Furthermore the *AuthorizedRole*, which
1106 participates in the *BusinessRealization* must be the source of exactly one *mapsTo* dependency
1107 targeting an *AuthorizedRole* participating in a *BusinessCollaborationUseCase*.

1108 C.98. *AuthorizedRoles* in a *BusinessRealizationView* must have a unique name within the scope of
1109 the package, they are located in

C.99. The number of *AuthorizedRoles* participating in a *BusinessCollaborationUseCase* MUST match the number of *AuthorizedRoles* participating in the *BusinessRealization* realizing this *BusinessCollaborationUseCase*

5.2.4.5 Example (informative)

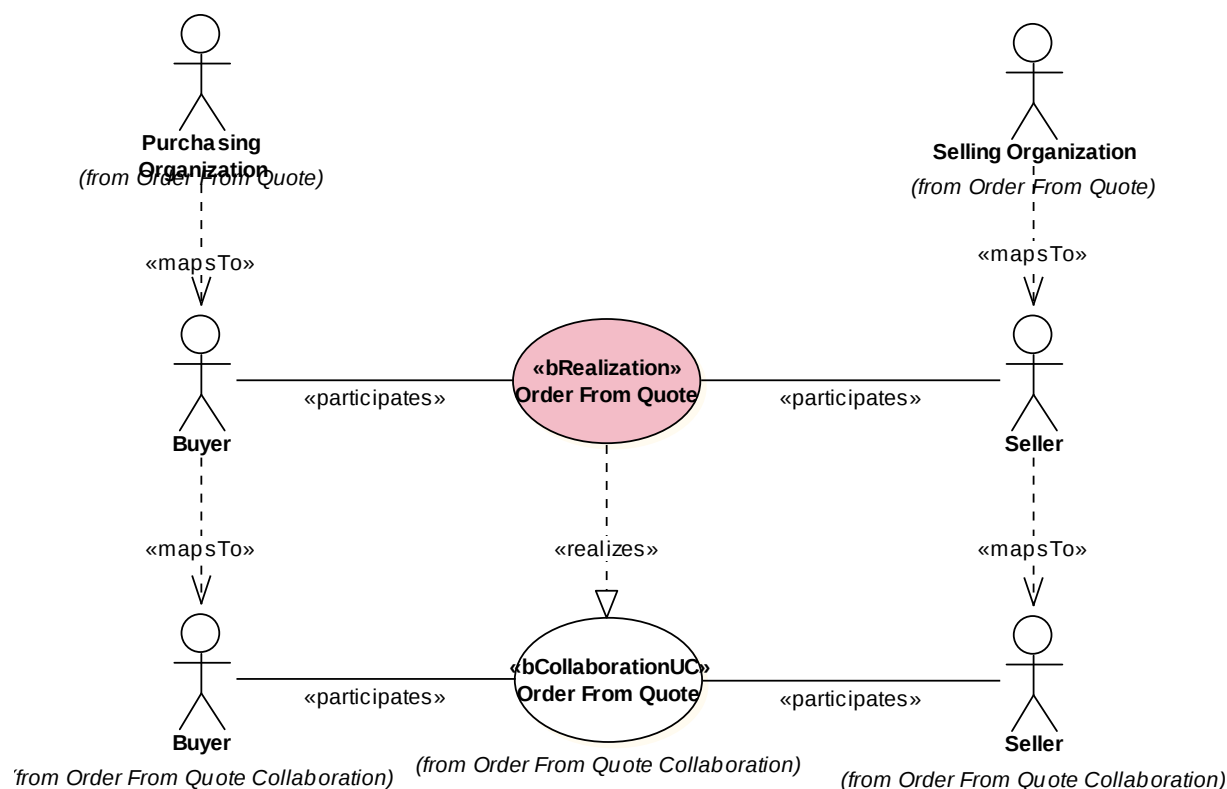


Figure 46 BusinessRealizationView - Example: Realization of the OrderFromQuote Collaboration between Purchasing Organization and SellingOrganization

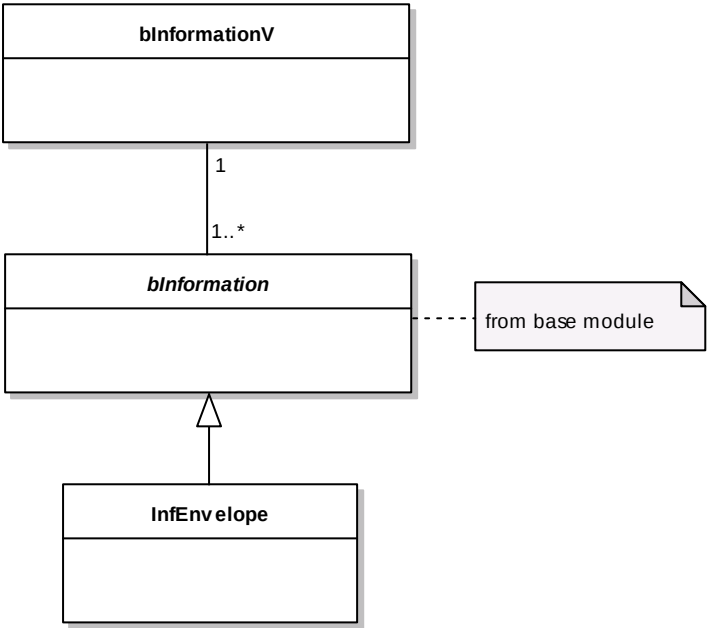
1119 **5.3 Business Information View**

1120 **5.3.1 Abbreviations of Stereotypes**

Stereotype Abbreviation	Full Stereotype Name
bInformationV	BusinessInformationView
bInformation	BusinessInformation
InfEnvelope	Information Envelope

1121

1122 **5.3.2 Conceptual Description (informative)**



1123

1124 **Figure 47 BusinessInformationView - Conceptual Overview**

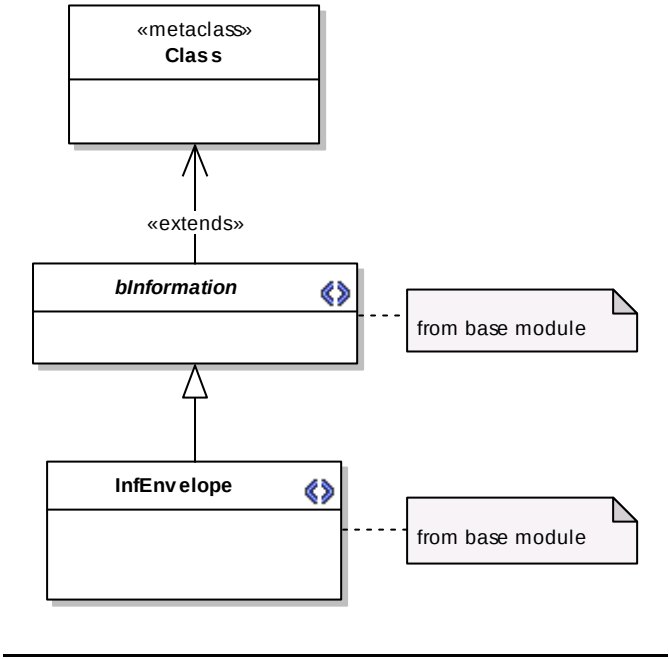
1125 A *BusinessInformationView* is a container of artifacts that describe the information exchanged in a
1126 *BusinessTransaction*. As previously mentioned; *RequestingInformationPin* and
1127 *RespondingInformationPin* are classified by an *InformationEnvelope* which is a subclass of a
1128 *BusinessInformation*. A *BusinessInformation* serves as an abstract container for all of the information
1129 exchanged between the *RequestingAction* and the *RespondingAction* or vice versa, respectively. The
1130 stereotypes *BusinessInformation* and *InformationEnvelope* is part of the UMM base module and
1131 imported into the UMM foundation module.

1132 The current UMM foundation module does not mandate a specific business information modeling
1133 approach. All methodologies and rules to build quality class diagrams can be used in order to model the
1134 exchanged information, as long as the root element of the data structure generalizes the
1135 *InformationEnvelope* class being part of the UMM base module.

1136 However, UMM strongly suggests using UN/CEFACT’s Core Components and Core Components Message
1137 Assembly artifacts to model the business information. Because Core Components are syntax
1138 independent and stereotyped, the usage of the UML Profile for Core Components is suggested within
1139 the *BusinessInformationView*.

1140

1141 5.3.3 Stereotypes and Tag Definitions (normative)



1142
1143
1144 Figure 48 BusinessInformationView - Abstract Syntax
1145

Stereotype	bInformation (BusinessInformation)
Base Class	Class
Parent	BusinessInformationView
Description	A <i>BusinessInformation</i> realizes abstract business document information that is exchanged between authorized roles performing activities in a business transaction. Since a <i>BusinessInformation</i> is defined as abstract it cannot be used directly in order to set the type of exchanged information in a <i>BusinessInformation</i> . Instead the concept of an <i>InformationEnvelope</i> is used.

1146

Stereotype	InfEnvelope (InformationEnvelope)
Base Class	Class
Parent	BusinessInformationView
Description	An <i>InformationEnvelope</i> is a subtype of a <i>BusinessInformation</i> and represents a concrete business message which is exchanged in a UMM business transaction. Any business document artifacts are connected to an <i>InformationEnvelope</i> using associations.

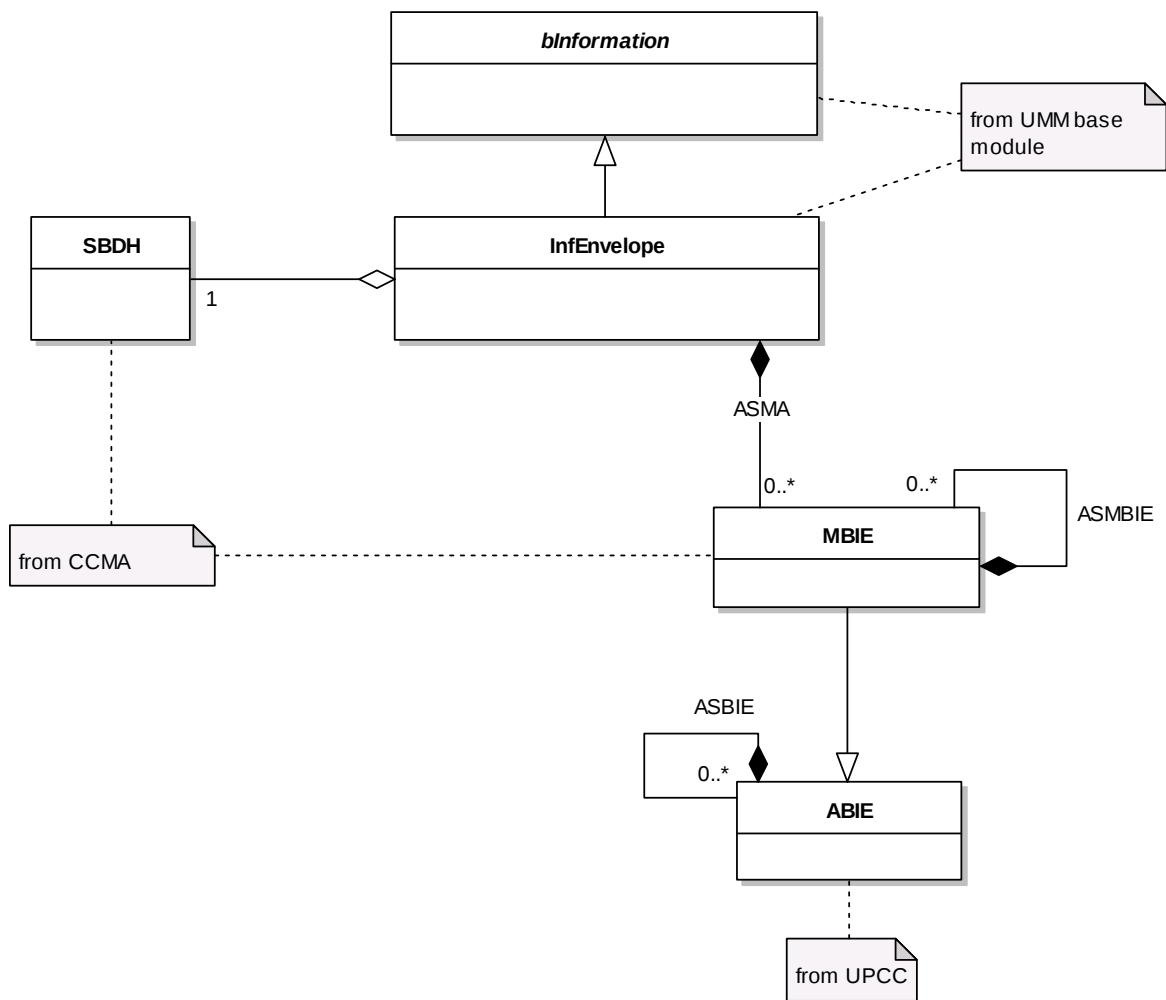
1147

1148

1149 5.3.4 Constraints (normative)

1150 C.100. A *BusinessInformationView* MUST contain one to many *InformationEnvelopes* or
1151 subtypes thereof defined in any other extension/specialization module. Furthermore, it MAY
1152 contains any other document modeling artifacts.

1153 **5.3.5 Example using UPCC (UML Profile for Core Components) and CCMA (Core**
 1154 **Components Message Assembly) (informative)**
 1155 The following example shows how to model the information exchanged in a *BusinessTransaction* using
 1156 the UML Profile for Core Components (UPCC) and the Core Component Message Assembly
 1157 (CCMA). Figure 49 shows how UMM, UPCC and CCMA are related to each other.



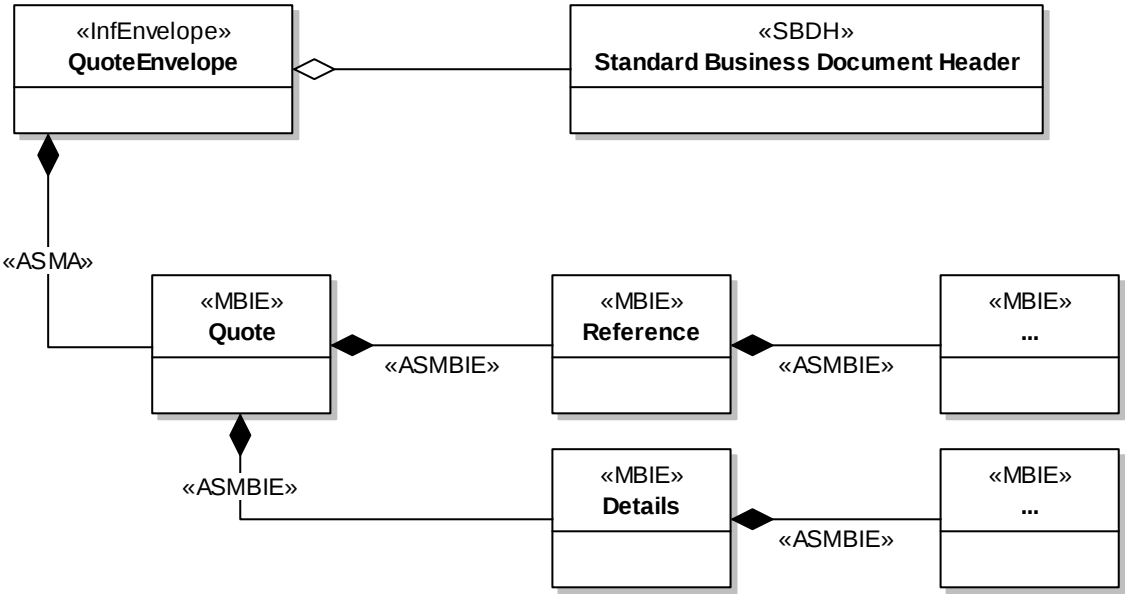
1158
 1159 **Figure 49 Abstract example for using UPCC artifacts to model business information**

1160 An *information envelope* has a *standard business document header* (SBDH) which serves for
 1161 identification purposes of technical sender and receiver, document type etc.

1162 The body of an *information envelope* consists of zero or more elements – so called *messaging business*
 1163 *information entities* (MBIE). MBIEs are connected to an *information envelope* using so called *association*
 1164 *message assemblies* (ASMA). If different *messaging business information entities* are connected to each
 1165 other the concept of *association messageing business information entities* (ASMBIE) is used. A
 1166 *messaging business information entity* inherits from an *aggregate business information entity* (ABIE).
 1167 ABIEs are part of the UML Profile for Core Components (UPCC) standard. In order to relate different
 1168 *aggregate business information entities* to each other so called *association business information entities*
 1169 (ASBIE) are used.

1170 *Standard business document header and messaging business information entities* are part of the core
1171 components message assembly (CCMA). *Aggregate business information entities* are part of the UML
1172 Profile for Core Components (UPCC).

1173 Figure 50 shows an example for a QuoteEnvelope modelled using concepts from the Core Components
1174 Message Assembly standard. Not shown in Figure 50 are the dependencies between ABIEs and MBIEs.



1175
1176 **Figure 50 CCMA example**

1177 Copyright Statement

1178

1179 Copyright © UN/CEFACT 2008. All Rights Reserved.

1180 This document and translations of it may be copied and furnished to others, and derivative works that
1181 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
1182 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
1183 notice and this paragraph are included on all such copies and derivative works. However, this document
1184 itself may not be modified in any way, such as by removing the copyright notice or references to
1185 UN/CEFACT except as required to translate it into languages other than English.

1186 The limited permissions granted above are perpetual and will not be revoked by UN/CEFACT or its
1187 successors or assigns.

1188 This document and the information contained herein is provided on an "AS IS" basis and UN/CEFACT
1189 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY
1190 THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1191 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.