



---

Creating A Single Global Electronic Market

## **ebXML Glossary**

### **Technical Coordination and Support Project Team**

**Date:** July 19, 2000

**This version:** 1.0

**Last version:** 0.3

**Status:** This document is for public comment

## Table Of Contents

Table Of Contents .....	2
Document Purpose.....	5
Introduction.....	5
Conventions.....	5
Glossary .....	6
Actor (BPM) .....	6
Administrator (TA) .....	6
Agent. (BPM).....	6
Application. (TRP) .....	6
Atomic Particle. (TA).....	6
Business. (REQ).....	6
Business Activity. (BPM).....	6
Business Document. (BPM).....	7
Business Process. (BPM) .....	7
Business Process Interface. (BPM) .....	7
Business Process Model. (REQ) .....	7
Business Process Modeling Specification (REQ) .....	7
Business Rule. (BPM) .....	7
Business Signal. (BPM).....	7
Business Transaction. (BPM) .....	8
Choreography (BPM).....	8
Class. (CC).....	8
Common Business Definition. (REQ) .....	8
Common Business Process. (REQ) .....	8
Common Character Encoding. (REQ) .....	8
Common Expression. (REQ) .....	8
Common Item. (REQ).....	9
Common Language. (REQ).....	9
Common Semantics. (REQ) .....	9
Community. (BPM) .....	9
Context. (REQ).....	9
Context categories (CC) .....	9
Contract. (BPM).....	9
Contract type. (BPM) .....	10
Core Component (CC).....	10
Core Components Specification (REQ) .....	10
Data element (TA) .....	10
Dictionary. (BPM) .....	10
Digital Signature (TRP).....	10
Document (TRP).....	10
Document Envelope. (BPM) .....	11
Document Exchange (TRP).....	11

Duality. (BPM) .....	12
ebXML Information Services. (R&R).....	12
Economic Event. (BPM) .....	12
Economic Resource. (BPM) .....	12
Economic Resource Type. (BPM) .....	12
Entity (CC).....	13
Entity ID (CC) .....	13
Entity Name (CC) .....	13
Fundamental Information Entity. (BPM).....	13
Hierarchical Context Classifications. (CC).....	13
Information Entity. (BPM).....	13
Information Flow. (BPM) .....	13
Logging Services. (R&R) .....	14
Market. (BPM) .....	14
Message (TRP) .....	14
Message Envelope (TRP).....	14
Message Types (TRP).....	14
Message Header (TRP).....	15
Message Manifest (TRP).....	15
Message Routing Information (TRP) .....	16
Metamodel (BPM).....	16
Object (TMWG) .....	16
Object Class(TMWG) .....	16
Partner. (BPM).....	16
Partner Type. (BPM).....	17
Partner Role. (BPM) .....	17
Party. (BPM) .....	17
Party (TRP).....	17
Party role. (BPM) .....	17
Party type. (BPM) .....	18
Pattern (CC) .....	18
Pattern ID (CC).....	18
Pattern Name (CC).....	18
Process Category. (BPM) .....	18
Quality Assurance Services. (R&R).....	18
Query Services. (R&R) .....	18
Registry (TA) .....	18
Registry and Repository Specification (REQ).....	19
Repository (TA) .....	19
Repository Interface Discovery Services. (R&R) .....	19
Repository Object (TA) .....	19
Resource Catalog. (BPM).....	20
Scenario (TA) .....	20
Service. (BPM) .....	20
Service. (TRP).....	20
Service Choreography. (TRP) .....	20

Step. (BPM) .....	21
Steward / Owner (TA) .....	21
Sub-class (CC) .....	21
Sub-Service. (TRP) .....	21
Technical Architecture Specification (REQ).....	22
Transaction (TRP) .....	22
Transformation Services. (R&R).....	23
Transport, Routing and Packaging Specification (REQ).....	23
WorkflowServices. (R&R) .....	23
Acronyms.....	24

# 1 Document Purpose

## 2 Introduction

3 This document has been prepared to assist in the co-ordination of specifications  
4 produced by the ebXML initiative.

5

6 It attempts to cross-reference common terms and indicate synonyms and  
7 homonyms where identified.

8

9 It should be used as the basis for technical assurance within the ebXML  
10 Technical Specification documentation.

11

12 It is understood that each Technical Specification document will use its own  
13 glossary of terms. However, these should defer to this glossary if an appropriate  
14 terms already exists herein.

15

16 A recent addition is the inclusion of an Acronym expansion section.

17

18 Please e-mail comments to Tim McGrath, [tmcgrath@tedis.com.au](mailto:tmcgrath@tedis.com.au).

19

20 This document is a working document and all comments are welcome. When  
21 responding with your comment, please provide the following information:

- 22 ➤ Your name,
- 23 ➤ Your email address,
- 24 ➤ The document page and line number(s) associated with your comment,
- 25 ➤ Your comment,
- 26 ➤ Rationale for the comment, and
- 27 ➤ Your recommended action for resolution of the issue or any recommended  
28 document add/change/delete modifications.

29

## 30 Conventions

31 Each term is listed alphabetically in ***Bold italicised*** typeface, followed by a code  
32 indicating the team which has attempted a definition (codes are defined in the  
33 Acronym section). For example...

34

35 ***term(TA)***

36 A definition of the term.

37

38 *{comments are in italics}*.

39

40 The goal is to add more structure to these definitions in the next release.

41

41

## 42 **Glossary**

### 43 ***Actor (BPM)***

44 A human or mechanical resource that can do work in the context of a business  
45 process.

46

### 47 ***Administrator (TA)***

48 The entity (person, organization, company) who operates a Registry or  
49 Repository. This designation is necessary to assign security responsibilities.

50

### 51 ***Agent (BPM)***

52 An agent is a particular kind of business process interface that represents an  
53 individual.

54

### 55 ***Application (TRP)***

56 An *Application* is software that may implement a *Service* by processing one or  
57 more of the *Messages* in the *Document Exchanges* associated with the *Service*.

58

### 59 ***Atomic Particle (TA)***

60 Representation classes.

61

### 62 ***Attribute (OMG)***

63 A conceptual notion. An attribute of an object is an identifiable association  
64 between the object and some other entity or entities.

65

### 66 ***Business (REQ)***

67 The term business is to be interpreted as interchangeable with for-profit, non-  
68 profit, not-for profit, and government entities.

69

### 70 ***Business Activity (BPM)***

71 A business activity is used to represent the state of the business process of one  
72 of the partners. For instance the requester is either in the state of sending the  
73 request, in the state of waiting for the response, or in the state of receiving (and  
74 processing) the response.

75

76 ***Business Document (BPM)***

77 A business document is the description of a particular entity within a business, or  
78 the description of an agreement between organizations, or the description of an  
79 business event. So the document is never the 'real' thing, just a description of it.  
80 A business document is the central component of any information exchange  
81 among partner roles.  
82

83 ***Business Process (BPM)***

84 A business process is a collection of actions and interactions that are required to  
85 achieve a business objective. Such objectives often are the execution of a  
86 business transaction or a related set of business transactions. Business  
87 processes can be decomposed to arbitrary levels of sub-processes.

88

89 ***Business Process Interface (BPM)***

90 A business process interface is the definition of how to interact with one partner  
91 role in order to make him/her perform a desired service. For instance a partner  
92 role can expose a business process interface for 'quotation service'. It will  
93 describe precisely what kind of business signal (i.e. message) you need to send,  
94 what you will get back, and what you may expect to have happen as a result of  
95 the exchange.

96 ***Business Process Model (REQ)***

97 Business process models will help define common items and provide their  
98 context.  
99

100 ***Business Process Modeling Specification (REQ)***

101 The business process metamodel and the recommended methodology for using  
102 it.  
103

104 ***Business Rule (BPM)***

105 A business rule is a very generic term to describe rules that govern how we  
106 conduct business. In this context a business rule is a rule that guides and  
107 constrains the execution of steps within a business process.  
108

109 ***Business Signal (BPM)***

110 A business signal is a message sent between the business process interfaces of  
111 two partner roles. A business signal fulfills the information flow requirements  
112 between request activity and response activity. A business signal contains  
113 business documents(s).  
114

115 ***Business Transaction (BPM)***

116 A business transaction is a logical unit of business conducted by two or more  
117 parties. The community, the partners, and the process, are all in a definable, and  
118 self-reliant state prior to the business transaction, and in a new definable, and  
119 self-reliant state after the business transaction. In other words if you are still  
120 'waiting' for your business partner's response or reaction, the business  
121 transaction has not completed.

122 A business transaction in our model is reflected as the required exchange or  
123 series of exchanges of information between two (or more) partner roles in order  
124 to complete the transaction. For example the exchange could consist of a  
125 request for quote and the return either of the actual quote, or of the confirmation  
126 that the request had been received. It would not make sense to have the  
127 transaction (interaction) consist of the request only.

128 ***Choreography (BPM)***

129 *{awaiting definition}*

130

131 ***Class (CC)***

132 (eg. address)

133

134 ***Common Business Definition (REQ)***

135 Business process definition (BPDS)enable an organization to express its  
136 business processes so that they are understandable by other organizations,  
137 thereby enabling integration of business processes.

138

139 ***Common Business Process (REQ)***

140 Both entities involved in the exchange of data must be engaged in executing the  
141 same transaction in the context of a business process .

142

143 ***Common Character Encoding (REQ)***

144 UNICODE, which is specified in the W3C XML Version 1.0 technical  
145 specification, provides this.

146

147 ***Common Expression (REQ)***

148 Common set of XML element names, attributes and common usage of those  
149 attributes, common approach to document structure

150

151 ***Common Item (REQ)***

152 Common items are semantic units at any level that stay consistent across  
153 contexts, and therefore are reusable both within and between business exchange  
154 messages.

155

156 ***Common Language (REQ)***

157 Common vocabulary, with a one to one correspondence between words and  
158 meaning

159

160 ***Common Semantics (REQ)***

161 Common meaning, as distinct from words, expression, or presentation.

162

163 ***Community (BPM)***

164 A community is a collection of parties that have formed a set of mutual  
165 partnerships in support of a shared goal. Within a community a party takes on a  
166 particular role, and is now distinguished as being a 'partner' as opposed to just a  
167 'party'. Communities often, but not always, form as subsets of markets. What  
168 communities have in common are shared interests and shared processes.

169 Examples of communities are: A given company's entire supply chain, An  
170 alliance or joint venture of a number of companies to collaborate to offer  
171 complete solutions, A company and all its customers.

172

173 ***Context (REQ)***

174 Business process models will help define common items and provide their  
175 context. This context will in turn define the precise use of common items in  
176 messages exchanged among parties. ebXML should describe these items in  
177 terms such as UML artifacts that are independent of implementation syntax.

178 Factors (or attribute) that influences rules. Describing variation in the data  
179 structures and providing semantic refinement of a core component (eg. trading  
180 region/country).

181

182 ***Context categories (CC)***

183 (eg. Industry (eg automotive))

184

185 ***Contract (BPM)***

186 A contract is a mutual commitment that some future actual economic events will  
187 be executed. For instance, a sales order can be seen as a contract representing  
188 the commitment to ship and the reciprocal commitment to pay.

189 Contracts can have recursive relationships with other contracts, for example  
190 yearly contracts with monthly releases and weekly or daily shipping schedules.  
191 Contracts are fulfilled by the execution of the committed Economic Events.  
192

193 ***Contract type (BPM)***

194 A Contract Type is the abstract classification or definition of a Contract.  
195 Examples might be service contracts, orders, and committed-plans.  
196 As in other Type objects, Contract Types are not just categories, they can also  
197 define the rules and processes governing contracts of the type.

198

199 ***Core Component (CC)***

200 *{awaiting definition}*

201

202 ***Core Components Specification (REQ)***

203 The set of ebXML core components, or the prescribed methodology for deriving  
204 them.

205 ***Data element (TA)***

206 The same as a repository object (see above). A data element is presumed to be  
207 atomic in structure unless specifically named as a "Compound Data Element"

208

209 ***Dictionary (BPM)***

210 (Consider how core components organize terms in the dictionary for optimal re-  
211 use.)

212 The dictionary should contain data types, re-usable components, and the  
213 templates (DTD's) of the business documents, but not the documents  
214 themselves.

215 ***Digital Signature (TRP)***

216 A *Digital Signature* is a cryptographic signature over data contained in a  
217 *Message*, or elsewhere that are addressable via [URI]s, that permits the  
218 authenticity of the signer of the data to be determined, and helps detect if the  
219 data in the *Message* has changed.

220

221 ***Document (TRP)***

222 A Document is any data that can be represented in a digital form. Examples of  
223 Documents include:

- 224 • a set of XML Elements
- 225 • an XML Document
- 226 • an HTML Document

- 227 • a word processing file
- 228 • an Adobe Acrobat PDF file
- 229 • a binary file
- 230 • part of larger document.
- 231

232 ***Document Envelope (BPM)***

233 A document envelope is the wrapper of an information flow between partner  
234 roles.  
235 It is not completely within the scope of the business process project team, rather  
236 it belongs to the transport project team, but we need to all have a common  
237 agreement on the business aspects and core component aspects of what goes  
238 in the envelope.  
239

240 ***Document Exchange (TRP)***

241 A Document Exchange is a generic term for either a:

242 *One-Way Document Exchange* consists of a *One-Way Message* sent from one  
243 Party to a second Party, followed by an optional *Acknowledgement Message*  
244 sent by the second party back to the first party, followed by an optional *Error*  
245 *Message* if an error was detected in the *One-Way Message*.

246 Examples of a *One-Way Document Exchange* include a supplier sending catalog  
247 updates to their buyers  
248

249 A *Simple Document Exchange* consists of a *Request Message* sent from one  
250 Party to a second Party, followed by an optional *Acknowledgement Message*  
251 sent by the second party back to the first party, followed by either:

252 an optional *Error Message* if an error was detected in the *Request Message*, or  
253 an optional *Checked OK Message*, if no errors were detected

254 that is sent by the second party back to the first party followed by a *Response*  
255 *Message* that is returned as a result of processing the *Request Message*.

256

257 Examples of instances of a *Simple Document Exchange* include:

258 a Purchase Order sent by a buyer to a seller and the acknowledgement from the  
259 seller of its receipt

260 a Purchase Order sent by a buyer to a seller and the Invoice that is sent back as  
261 a result of fulfilling the order

262 sending a document for review by a lawyer followed by the legal opinion that is  
263 sent back as a result

264

265 A *Multiple Round Trip Document Exchange* consists of a Request Message sent  
266 from one Party to a second Party, followed by a series of *Exchange Messages*  
267 that are exchanged between the two *Parties* until finally either the first or the  
268 second *Party* generates and sends a *Response Message* back to the other  
269 *Party*.

270 Examples of *Multiple Round Trip Document Exchanges* include:

271 the exchange of messages required to make a payment using payment method  
272 protocols such as [SET] or [Mondex]

273 the exchange of messages required to negotiate an agreement on terms and  
274 conditions.

275

### 276 ***Duality (BPM)***

277 Duality is a relationship between Economic Events, where one is the legal or  
278 economic consideration of the other. Examples include a payment for a product  
279 or service.

280

### 281 ***ebXML Information Services (R&R)***

282 • archives of previous ebXML technical specifications

283 • online access requirements of the other ebXML project teams as defined by  
284 their requirements and deliverables

### 285 ***Economic Event (BPM)***

286 An Economic Event is the transfer of control of an Economic Resource from one  
287 party to another party. Examples would include sale, cash-payment, shipment,  
288 and lease.

289

### 290 ***Economic Resource (BPM)***

291 An Economic Resource is a quantity of something of value that is under the  
292 control of an enterprise. Examples: cash, inventory, and labor and machine  
293 services.

294

### 295 ***Economic Resource Type (BPM)***

296 An Economic Resource Type is the abstract classification or definition of an  
297 Economic Resource. For example, in an ERP system, ItemMaster or  
298 ProductMaster would represent the Economic Resource Type that abstractly  
299 defines an Inventory Item or Product.

300 Economic Resource Types could have recursive relationships, so that for  
301 example broad classifications like "product" could group smaller classifications

302 like "product family", which could have as members specific "product masters"  
303 with SKU numbers.  
304 Economic Resource Types are not just categories; they may be complex objects  
305 that define complex information for Economic Resources such as bills of  
306 material, prices, etc.  
307 (Note: In order to tie this into the search for a product or a service, the  
308 metamodel needs to allow parties or partners to offer economic resource types.)  
309

310 **Entity (CC)**

311 {awaiting definition}

312

313 **Entity ID (CC)**

314 {awaiting definition}

315

316 **Entity Name (CC)**

317 {awaiting definition}

318

319 **Fundamental Information Entity (BPM)**

320 A Fundamental Information Entity is in essence a data type. In business contexts  
321 we might need many more 'data types' with business semantics beyond the  
322 standard data types of 'int', 'float' etc.

323

324 **Hierarchical Context Classifications (CC)**

325 {awaiting definition}

326

327 **Information Entity (BPM)**

328 An information entity is a primitive or complex data structure. We haven't defined  
329 this yet, but it may be that the difference between a data structure and an  
330 information entity is that the information entity also contains business rules about  
331 the data.

332

333 **Information Flow (BPM)**

334 An information flow is a flow of information between partner roles, related to a  
335 specific set of business activities within a specific business transaction. Often the  
336 information flow will specify a particular business document to be exchanged  
337 between the partners before the interaction or the general process can proceed.

338 **Logging Services (R&R)**

339 The ability to the ability the store transactional and query events and metrics.

340

341 **Market (BPM)**

342 A market is a 'meeting place' where organizations and individuals can exchange  
343 services or products. A market is defined in terms of the types of services and  
344 products that are likely to be exchanged. The "Yellow Pages" in our phonebook  
345 is a good example of classifications of products and services, e.g. 'Legal  
346 Services', or 'Air condition products'. So you can then anticipate the existence of  
347 a 'Legal Services' market and a 'Air Conditioning' market.

348

349 **Message (TRP)**

350 A Message is data that is sent from one Party to another.

351 All the data in a Message is contained within a Message Envelope.

352 A Message consists of a Message Header and a Message Body

353 Examples of a Message include:

- 354 • a Purchase Order that is sent by a buyer to a supplier
- 355 • an Invoice that is sent by the supplier back to the buyer
- 356 • a request to make a payment of \$50 sent to a Credit Card acquirer
- 357 • the authorization received from a Credit Card acquirer as a result of making a  
358 payment
- 359 • Status indicating the success or failure of a Service

360 **Message Envelope (TRP)**

361 A *Message Envelope* is the outermost container for a Message. It can be such  
362 things as:

363 an XML Document, or

364 a multi-part MIME message

365 **Message Types (TRP)**

366 Messages may be of several different types. These are described below.

- 367 • A *One-Way Message* is a *Message* sent from one party to another. The  
368 receiving *Party* MAY only respond back to the From Party with either a  
369 *Message Acknowledgement* and, if there is an error, an *Error Message*.
- 370 • A *Request Message* is a *Message* sent from one *Party* to a another *Party's*  
371 *Service* with the intent that the other *Party* act upon the data in the *Request*  
372 *Message* by carrying out the *Service*. The results of processing the *Request*  
373 *Message* MUST be included in a *Response Message* that is sent back to the  
374 sender of the previous *Message*.
- 375 • An *Acknowledgement Message* may sent as a response to any *Message*  
376 (apart from an *Acknowledgement Message*) to indicate that the Message has  
377 been received.

- 378 • A *Checked OK Message* may be sent in response to a *Request Message* to  
379 indicate that the content of the *Request Message* has been validated and no  
380 errors were found. A *Checked OK Message* MUST be sent after any  
381 *Acknowledgement Message* that was sent.
- 382 • A *Response Message* is a *Message* that is generated by the *Service* that  
383 received a *Request Message*. It is produced as a result of carrying out the  
384 requested *Service*. It is the last *Message* in a *Document Exchange* unless the  
385 *Response Message* contains errors. *Response Messages* are sent back to  
386 the sender of the *Request Message*.
- 387 • An *Exchange Message* is a *Message* that is sent between one *Party* and  
388 another after the sending of the initial *Request Message* and before the  
389 sending of the final *Response Message*. Examples of *Exchange Messages*  
390 include:
- 391 intermediate messages that are part of a Payment Protocol  
392 a counter offer to an offer made as part of a negotiation.
- 393 • An *Error Message* is a *Message* that reports on a problem in an earlier  
394 *Message* that prevents the earlier *Message* from being processed in a normal  
395 way. Examples of an *Error Message* include:
- 396 an *Error Message* reporting that an XML document was invalid or did not  
397 conform to its XML schema
- 398 an *Error Message* reporting a Transient Error that the Server processing a  
399 *Message* is busy and therefore the original *Message* should be resent at a later  
400 point in time
- 401 an *Error Message* that reports on an error in the underlying transport protocol.  
402
- 403 ***Message Header (TRP)***
- 404 A *Message Header* is an XML construct that contains the additional data that  
405 needs to be associated with the *Documents* in a *message* so that they can be  
406 sent to and successfully processed by a *Party*.
- 407 ***Message Manifest (TRP)***
- 408 The *Message Manifest* contains references to the other documents, apart from  
409 the *Message Routing Information* document, that are contained within the same  
410 *Message Envelope*.
- 411 The purpose of the *Message Manifest* is to facilitate locating and validating that  
412 all required *Documents* contained within the *Message Envelope* are present.  
413 Examples of the types of documents that might be referenced by a *Message*  
414 *Manifest* include:
- 415 a *Purchase Order*
- 416 a *Purchase Order* and a picture of the requested goods
- 417 a *Purchase Order* and a digital signature

418

419 **Message Routing Information (TRP)**

420 *Message Routing Information* contains data that indicates the path that should be  
421 or was taken by a *Message* in reaching its ultimate destination.

422

423 **Metamodel (BPM)**

424 {*awaiting definition*}

425

426 **Object (TMWG)**

427 An object is a unique and specific instance or occurrence within an object class.  
428 As such, it is an identified, integral software component of a business activity or  
429 system such as a customer account, branch, transaction, etc. in a computerized  
430 banking system. It contains encapsulated computer language which describes its  
431 content (i.e., what it is), its behavior (i.e., what it is capable of doing in terms of  
432 incoming and outgoing messages or instructions), and its method, (i.e., how it  
433 interfaces with other objects in order to perform its intended purposes). Once  
434 identified and defined in a computer language, objects become self-contained  
435 and can be reused in multiple computer applications with little or no modification.

436 Techniques and Methodologies Working Group Information

437 <http://tmwg.harbinger.com/faq.html#t7>

438 **Object Class(TMWG)**

439 An object class is a group of logically related instances, sometimes termed a  
440 domain, which have common behavioral characteristics and other significant  
441 similarities. For example, a single customer may be identified as an object  
442 whereas all customers would be considered an object class. Object classes may  
443 be further subdivided into subclasses such as a group of customers within a  
444 given geographical area or sharing a specific buying pattern. The correct  
445 grouping of objects into classes is very important, as software developers will  
446 generally base programs only on object classes, not individual objects.

447 <http://tmwg.harbinger.com/faq.html#t7>

448

449 **Partner (BPM)**

450 A partner is a participant in a community. It is defined in terms of it's generic  
451 partner type and its particular partner role(s) within processes within the  
452 community.

453

454 ***Partner Type (BPM)***

455 A partner type is a broad classification of the kind of role an organization or  
456 individual is able to play within a community. Examples would be 'supplier',  
457 'agent', 'consultant', 'administrator', 'consumer'.  
458 (Note: Consider removing the inheritance between party type and partner type)  
459

460 ***Partner Role (BPM)***

461 A partner role is an active engagement in a process within a community.  
462 A partner role is always related to a process, and typically describes the legal or  
463 economic responsibility in one or more business transactions. Examples are  
464 'buyer', 'seller', 'scheduler', 'dispatcher', 'approver', 'pricer'.  
465 (Note: Consider removing the inheritance between party role and partner role)  
466

467 ***Party (BPM)***

468 A party is any organization or individual that needs to participate in exchange of  
469 products or services in one or more markets. A party is established first as an  
470 absolute entity and then in terms of the roles it plays in a market.

471 ***Party (TRP)***

472 A Party is a company, organization or individual or other entity that can generate,  
473 receive or relay Documents.

474 Examples of a Party include:

- 475 • a Merchant
- 476 a Customer
- 477 • a Lawyer
- 478 • a Bank
- 479 • a government department or agency
- 480 • an intermediary or agent
- 481 • a software agent

482 A Party is also used to refer to systems or servers that are carrying out Services  
483 or processes on behalf of a Party.

484 ***Party (CC)***

485 Used as an example of a core component.

486

487 ***Party role (BPM)***

488 A party role is the role a party plays within a given market. Examples would be  
489 'manufacturer', 'whole-saler', 'consultancy', 'logistics'. A party can have one or  
490 more roles within each market.

491 (Note: If my interpretation is right our class diagram needs to reflect Party Role  
492 as a junction between market and party.)

493

494 **Party type (BPM)**

495 A party type is a broad classification of the kind of organization or individual.  
496 Examples would be 'University', 'Corporation', 'Individual', 'Government'.

497

498 **Pattern (CC)**

499 {awaiting definition}

500

501 **Pattern ID (CC)**

502 {awaiting definition}

503

504 **Pattern Name (CC)**

505 {awaiting definition}

506

507 **Process Category (BPM)**

508 A process category is a broad classification of business processes. At a macro  
509 level this classification could be like the "Yellow Pages" classification of services.  
510 At a finer level processes could be classified to more functional groupings such  
511 as 'quotation', 'scheduling',  
512 The metamodel does not constrain the kinds of classification of processes.

513 **Quality Assurance Services (R&R)**

514 The ability to validate content based on its classification.

515

516 **Query Services (R&R)**

517 The ability to send a request and retrieve results from a physical storage  
518 mechanism, e.g., exact or similar matches and navigation.

519 **Registry (TA)**

520 A Registry is a mechanism whereby relevant Repository Objects and metadata  
521 about them can be registered such that a pointer to their location, and all their  
522 meta data, can be retrieved as the result of a query.

523

524 A registry must also be able to track and recognize Owners of Repository  
525 Objects and the Repository Objects themselves (see "2.1.4 – Registry and  
526 Repository Security")

527

528 A Registry must incorporate a mechanism for querying a repository or a cache of  
529 a repository's index via an API.

530

531 A registry is required to allow process owners to submit, classify, register and  
532 update mapping templates, business process specifications, and data  
533 interchange specifications. This registry should have an Application Program  
534 Interface (API) expressed in XML which would also support human interfaces  
535 through manual HyperText Transfer Protocol (HTTP). This registry should  
536 support an agreed upon security protocol.  
537

538 ***Registry and Repository Specification (REQ)***

539 Includes functional specification and technical design, interfaces, services.  
540

541 ***Repository (TA)***

542 A location or a set of distributed locations where Repository Objects pointed at  
543 by the registry reside and from which they can be retrieved by conventional  
544 (http / ftp) means, perhaps with additional authentication/permission layers.  
545 Repositories must be globally distributed. Repositories can be maintained by top  
546 level organizations, verticals or SME's.

547 A repository is required for storage and retrieval of various items that support  
548 performing business electronically. There are two distinct sets of business  
549 requirements on the repository:

- 550 • a set dealing with managing the workflow of developing standard components  
551 that are stored in the repository, and
- 552 • a set dealing with application usage of the repository.

553 Additionally, the repository should support the information needs of the ebXML  
554 work group and project teams, as well as ebXML technical specification users  
555 with respect to glossaries and products.  
556

557 ***Repository Interface Discovery Services (R&R)***

558 The ability to expose (sub)set of ebXML interfaces implemented by a repository.

559

560 ***Repository Object (TA)***

561 A generic description of ANY object, whether it be a process or an actual Data  
562 Element, that is stored in a Repository. Repository Objects can be atomic or  
563 complex structures of atomic data elements.

564 ***Repository Object (RR)***

565 The data elements for the metadata associated with the submission, the  
566 submission itself ( i.e. the schema or the DTD ) has to be whole schema or DTD.

567

568 **Resource Catalog (BPM)**

569 A resource catalog is basically a navigable guide to offered products and  
570 services (Economic Resource Types). It is the market equivalence of a  
571 company's product catalog. It would be intended for narrowing down the  
572 particular kind of product or service you are looking for, hopefully leaving you  
573 with multiple possible sources for that product or service.

574 **Scenario (TA)**

575 {awaiting definition}

576

577 **Service (BPM)**

578 A service is a particular kind of business process interface that represents an  
579 organization.

580

581 **Service (TRP)**

582 A *Service* is a process that can be carried out by a *Party* as a result of receiving  
583 a *Request Message* or *One-Way Message* that requests the execution of that  
584 *Service*.

585 A *Service* can consist of either a Document Exchange or a set of Sub-Services.  
586 Examples of a *Service* include:

587 a Purchasing Service that enables a customer to purchase goods on-line

588 an Order Processing Service that processes an Order and generates a response  
589 as a result

590 a Payment Service that accepts a payment and provides a receipt

591 a Fulfillment Service that fulfills an order at the request of a Merchant.

592

593 **Service Choreography (TRP)**

594 A *Service Choreography* is a description of the dependencies that control the  
595 sequence and choices that determine which *Sub-Services* are executed when  
596 carrying out a *Transaction*.

597 The *Sub-Services* in a *Service* will have dependencies between them.

598 Dependencies can be:

599 *Serial*. One *Sub-Service* must start only after the completion of another *Sub-*  
600 *Service*

601 *Alternative*. One *Sub-Service* may be executed as an alternative to another

602 *Iterative Loop*. A *Sub-Service* may be repeated a variable number of times

603 *Conditional*. The execution of a *Sub-Service* is conditional on the state of another  
604 *Service*. This may be used in conjunction with *Serial*, *Alternative* and *Iterative*  
605 *Loop* dependencies.

606 *Parallel*. A *Sub-Service* may execute in parallel with another *Service*

607 *Concurrent*. A *Sub-Service* must execute at the same time as another *Sub-*  
608 *Service*.

609 An example of a simple *Sub-Service Choreography* is a Purchase Service that  
610 consists of three *Sub-Services*:

611 an Offer Service that conveys an Offer for sale of goods. This *Sub-Service* has  
612 no dependencies and therefore starts first

613 a Payment Service that carries out the Payment which has a Serial dependency  
614 on the Offer Service

615 a Delivery Service that delivers the Digital Goods, that has a Serial Dependency  
616 on the Payment Service

### 617 **Step (BPM)**

618 A step is a decomposition of a process that has a dependency on another  
619 decomposition. That dependence could be a predetermined sequence, or it could  
620 be otherwise determined through simple or complex business rules. A step is  
621 always either an action taken by a single partner role or an interaction among  
622 partner roles.  
623

### 624 **Steward / Owner (TA)**

625 A generic descriptor of a custodian who owns a Repository Object.  
626

### 627 **Sub-class (CC)**

628 (eg. billing address)  
629

### 630 **Sub-Service (TRP)**

631 A *Sub-Service* is a *Service* that is executed at the request of and as part of  
632 another *Service*.

633 Examples of *Sub-Services* include:

634 a payment service that occurs as part of a purchase

635 a tax calculation service that calculates the tax due as part of an order  
636 processing service.  
637

638 ***Technical Architecture Specification (REQ)***

639 Contains an overview of the technical infrastructure that comprises ebXML and  
640 itemize the design rules and guidelines.

641

642 ***Transaction (TRP)***

643 A *Transaction* is an instance of the execution of a *Service*.

644

645 Examples of a *Transaction* include:

646 a Purchase Transaction that buys a Company Report for \$20. It consists of three  
647 Sub-Service instances:

648 an Offer Service instance to buy the Company Report for \$20

649 a Payment Service instance that accepts a Payment for \$20 using a credit card,  
650 and finally

651 a Delivery Service instance that delivers the Company Report as an HTML web  
652 page.

653 a Buying Service that consists of the following Sub-Services:

654 three Price Negotiation Service instances that negotiate the price of a  
655 Photocopier.

656

657 There are several different meaning that have been associated with transactions:

658 "ACID" transactions (TBD) A transaction can be considered a collection of  
659 actions with the following properties:

660 Atomicity. A transaction's changes to the state are atomic: either all  
661 actions happen or none happen.

662 Consistency. A transaction is a correct transformation of the state. The  
663 actions taken as a whole do not violate any of the integrity constraints  
664 associated with the state. This requires that the transaction be a correct  
665 program.

666 Isolation. Even though transactions execute concurrently, it appears to  
667 each transaction T, that others executed either before or after T, but not  
668 both. In other words, each transaction is isolated from any others.

669 Durability Once a transaction completes successfully (commits), its  
670 changes to the state survive failures.

671 "EDI" transactions - "The information included in a transaction set is, for the most  
672 part, the same as the information in a conventionally printed document. A  
673 transaction set is the data that is exchanged in order to convey meaning between  
674 parties engaged in EDI"Conversational" transactions - A conversation is a  
675 sequence of related transactions between two parties separated in time. A  
676 complete "unit of business" for example, the negotiation of a purchase,  
677 placement, confirmation, payment and delivery of goods, may be represented as  
678 multiple transactions in a longer running conversation." From DISA publication  
679 titled "Introduction to EDI", (ASC X12S/94-190)

680 "Read-only" transactions - a transaction that consists of a document exchange  
681 where the information is obtained from a service without changing the state of the  
682 service  
683

684 ***Transformation Services (R&R)***

685 The ability to transform objects into another form. (e.g., IDEF-1X to XMI, XMI to  
686 XML Schema).  
687

688 ***Transport, Routing and Packaging Specification (REQ)***

689 Addresses transport of ebXML messages, the means of security employed, and  
690 the physical construction of the messaging used within the scope of the ebXML  
691 system.  
692

693 ***WorkflowServices (R&R)***

694 The ability to assign, route, sign-off, and define rules to support the workflow.

695

696

696

697 **Acronyms**

698

699 BOV Business Operational View

700

701 BPA[WG] Business Process Analysis [Working Group]

702

703 BPDS Business process definition

704

705 BPM ebXML project team: Business Process Methodology

706

707 CC ebXML project team: Core Components

708

709 CDWG/ITPWG Codes Working Group/International Trade Procedures Working  
710 Group

711

712 CEFACT Centre for the Facilitation of procedures and practices in  
713 Administration, Commerce and Transport

714

715 COM Component Object Model (Microsoft)

716

717 COM+ Forthcoming Microsoft distributed component architecture derived from  
718 DCOM and MTS

719

720 CORBA Common Object Request Broker Architecture (OMG distributed  
721 components)

722

723 CSG CEFACT Steering Group

724

725 CSS Cascading Style Sheets (XML)

726

727 DCD Document Content Description – extends XML DTD for enhanced content  
728 validation

729

730 DCOM Distributed Component Object Model (Microsoft distributed components)

731

732 DOM XML Document Object Model

733

734 DTD XML Document Type Definition

735

736 EBES European Board for EDI Standards

737

738 EJB Enterprise Java Beans (Java distributed components)

739

740 EMG EBES Modelling Group

741	
742	EWG-MT UN/EDIFACT Working Group – Management Team
743	
744	FSV Functional Service View
745	
746	IANA Internet Assigned Numbers Authority
747	
748	ISV Independent Software Vendor
749	
750	LWG Legal Working Group
751	
752	MEA ebXML project team: Marketing, Awareness, and Education
753	
754	MOF Meta Object Facility
755	
756	MOM Message Oriented Middleware
757	
758	MoU/MG Memorandum of Understanding Management Group
759	
760	MTS Microsoft Transaction Server
761	
762	OCL Object Constraint Language
763	
764	ODBC Open Database Connectivity
765	
766	OMG Object Management Group
767	
768	OO Object Oriented
769	
770	R&R ebXML project team: Registry and Repository
771	
772	RAD Rapid Application Development
773	
774	REQ ebXML project team: Requirements
775	
776	RMI Remote Method Invocation (Sun – used by Java Beans)
777	
778	SDO Standards Development Organisation
779	
780	SGML Standard Generalised Markup Language
781	
782	TA ebXML project team: Technical Architecture
783	
784	TCS ebXML project team: Technical Coordination and Support
785	
786	TRP ebXML project team: Transport, Routing, and Packaging

787  
788

