

1 ebXML Registry and Repository Part 1: Business Domain

2 *ebXML Working Draft issue date May 11, 2000*

3 Version: 1.0

4
5 Project Team Lead: Scott Nieman, Norstan Consulting, USA

6 Editor: Joe Dalman, TIE Commerce, USA

7 Contributors: Scott Hinkelman, IBM, USA; Terry Allen, CommerceOne, USA; Lisa Carnahan,

8 DOC/National Institute of Standards and Technology, USA; Eric Jarry, Software AG, France;

9 Kunio Mizoguchi, Ecom Japan; Jong L Kim, Inno Digital Co., Ltd, Korea; Michael Rowley,

10 Ph.D, eXcelon Corp, USA; Robert Miller, Global eXchange Services, USA; Sally Fuger, Ford

11 Motor Company, USA; Yutaka Yoshida, Sun Microsystems, USA; Nagwa Abdelgfour, Sun

12 Microsystems, USA; Gait Boxman, TIE Product Development, Holland

13 *Abstract*

14 This document is Part 1 of the Functional Requirement Specification for an ebXML-compliant
15 registry and repository. Part 1 defines the scope of the workflow to interact with a registry and
16 repository, the actors involved in these interactions and the overall Domain Architecture for
17 the e-Business Requirements of the registry and the repository. The Domain Architecture
18 provides the basis of the detailed workflow specifications as defined in *ebXML Registry and*
19 *Repository Part 2: e-Business Requirements* as well as forming the basis of traceability
20 between Part 1 and Part 2.

21 *Status of this Document*

22 This is the first public working draft of the *ebXML Registry and Repository Part 1: Business*
23 *Domain*, issued by the ebXML Registry and Repository Project Team for public review and by
24 members and project teams of ebXML.

25
26 This working draft incorporates the decisions of the ebXML Registry and Repository project
27 team as of May 11, 2000. The first comment period opens May 15, 2000 and ends May 29,
28 2000. Comments with associated starting line number and ending line number, and proposed
29 changes (required) should be emailed to joe.dalman@tiecommerceusa.com in DocBook, MS
30 Word or plain text format. We will not accept this document edited with revisions on.

31
32 During the first comment period, the project team will review public comments while
33 continuing its development of *ebXML Registry and Repository Part 2: e-Business*
34 *Requirements*. Part 2 will focus on the detailed workflows that highlight the business-to-
35 business (B2B) interchanges between users, a registry, and a repository.

36 *Table of Contents*

37	ebXML Registry and Repository Part 1: Business Domain	1
38	ebXML Working Draft issue date May 11, 2000.....	1
39	Abstract	1
40	Status of this Document	1
41	Table of Contents	1
42	1 Introduction	2
43	1.1 Purpose	2
44	1.2 Dependencies on other Specifications.....	3
45	1.3 Conventions and Terminology.....	4
46	2 Actor Relationships	4
47	2.1 Guest User	5
48	2.2 Corporation	5
49	2.3 Registration Authority	6
50	2.4 Industry Consortium	7
51	2.5 ebXML Business Application.....	8
52	3 Domain Use Case	9

53	3.1	Submitting Organization	9
54	3.2	Guest User	10
55	3.3	Independent Software Vendor (ISV)	11
56	3.4	Registration Authority (RA).....	11
57	3.5	ebXML-compliant Business Application (ebAppl)	12
58	3.6	Semantic Mapping Specialist	13
59	4	ebXML Registry and Repository Architecture (Domain Package Diagram)	13
60	4.1	Interrelationships between <<UI>>	14
61	4.2	Interrelationships between <<UI>> and <<Services>>	15
62	4.3	Interrelationships between <<Applications>> and <<Services>>	16
63	4.4	Interrelationships between <<ebXML-Arch>> and <<Services>>	17
64	4.5	Interrelationships between <<Services>>	18
65	4.6	Optional Services within the <<Service>> Registry package.....	19
66	4.6.1	<<Service>> Transformation Service	19
67	4.6.2	<<Service>> Workflow Service.....	19
68	4.6.3	<<Service>> Quality Assurance Service	19
69	4.7	Optional Services within the <<Service>> Repository package	20
70	4.7.1	<<Service>> Library Control Service	20
71		Appendix.....	20
72			

73 **1 Introduction**

74 This *ebXML Registry and Repository Part I: Business Domain* working draft defines the scope
75 and functional requirements of an ebXML-compliant registry and repository.

76
77 Section 2 defines the actors that can interact with a registry and repository, and represents
78 the relationships between these actors in an Actor Relationship diagram. This diagram is a
79 hierarchical view that depicts actor inheritance and specialization as well as dependencies
80 between actors.

81
82 Section 3 defines the scope of the registry and repository as reflected by a high-level
83 business domain Use Case. The use cases define the usage viewpoints from the perspective
84 of several actors, including that of an ebXML business application.

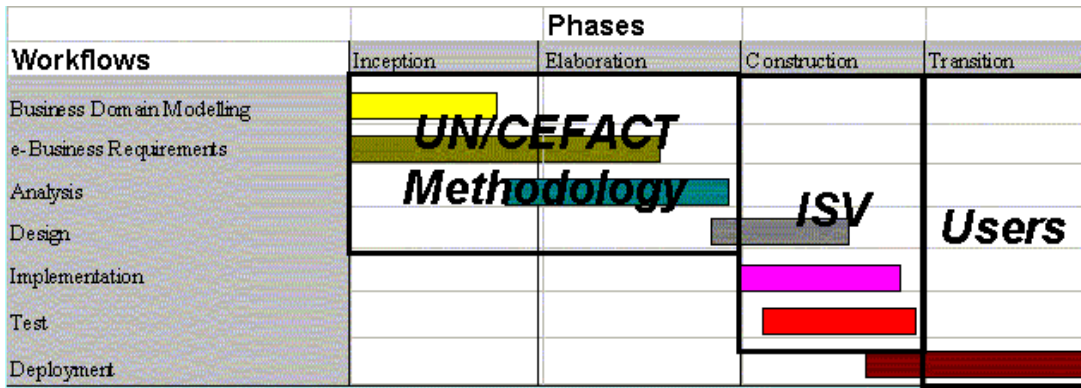
85
86 Section 4 defines the ebXML Registry and Repository overall architecture through the use of
87 a Domain Package diagram. Each package categorizes general functionality that could
88 potentially become the software components, but not necessarily, since some functionality
89 may be embedded within another service.

90 **1.1 Purpose**

91 The purpose of this working document is to define the overall system boundary as defined in
92 the UN/CEFACT/TMWG UML Profile and Methodology. It is to define the scope of the overall
93 system and provide a reasonable understanding of the size of the project.

94
95 The methodology covers four main workflows as shown below, as this working document is
96 the first of four documents.

97
98
99



100

101

Figure 1. Phases and Workflows

102

103

104

105

106

107

108

109

110

Ultimately, the result of the methodology will enable the ebXML Registry and Repository Project Team to apply Production Rules to generate the XML interfaces for both the Registry and the Repository. Software vendors can use the XML interfaces specifications as well as the UML models to understand the static and dynamic behavior of the system in order to construct the systems. With these specifications, conformance testing can be more clearly defined, since software components must conform to the XML interfaces, and the interactions to the system should be according to the UML activity and sequence diagrams.

111

112

113

114

115

116

117

118

It should be understood that this working document defines the scope for a network of registries and a network of repositories. This approach fits the model of world wide Internet, incorporating a vast number of interdependencies between companies, vertical domains relying on other vertical domains (finance, logistics, purchasing) and specifications relying on other specifications. It is envisioned that existing EDI directories could become repositories, and their traditional references to outside associations that maintain code lists (i.e., phone numbers, mailing addresses) could be replaced by ebXML based links from one repository to another.

119

120

121

122

123

124

125

While some organizations advocate the philosophy of a single global repository, ebXML Registry and Repository disagrees with that notion for the following reasons:

- 1) The approach will not scale
- 2) It does not fit the World Wide Web model
- 3) User associations, consortiums and standards organizations do not trust the notion of one single organization maintaining such a repository, deciding what should be in the repository and what should not be in the repository.

126

1.2 Dependencies on other Specifications

127

128

129

130

131

132

133

134

The Unified Modeling Language version 1.3, as issued by the Object Management Group, is used to define the entire ebXML Registry and Repository specification. The intent is a top-down model driven design approach versus a bottom-up approach based on data models or hand developed DTDs and brute force software development. This allows the business needs to drive the technology usage, provides a technology neutral representation of the specifications, and allows for the auto-generation of software code including XML specifications.

135

136

137

138

139

140

141

142

143

144

The UN/CEFACT/TMWG UML Profile and Methodology, which defines how and which UML artifacts need to be produced, is being utilized. It is adapted from the Rational Unified Process and specifically focused on e-Business Process modeling, including business-to-business (B2B) transactions. While its intent is to provide a protocol-neutral method of defining e-Business Processes, it also includes sample methods to auto-generate XML interfaces from the UML model by applying Production Rules. These XML interfaces follow a request/response model for real-time interaction with a registry and repository. Therefore the XML interfaces are intended to invoke object methods by passing XML requests and returning XML document responses.

145 EbXML Registry and Repository project team will consider any W3C Recommendations in the
 146 design or implementation of a ebXML-compliant registry and repository. W3C specifications
 147 in progress such as Xlink and Xpointer are key specifications that will be monitored. Such
 148 specifications should be helpful in enabling internetworking of registries to repositories and
 149 repositories to repositories.

150
 151 The Registry and Repository project team is considering transforming the Registry and
 152 Repository UML model to XML as according to the Object Management Group XML Metadata
 153 Interchange (XMI) v1.1 specification.

154 **1.3 Conventions and Terminology**

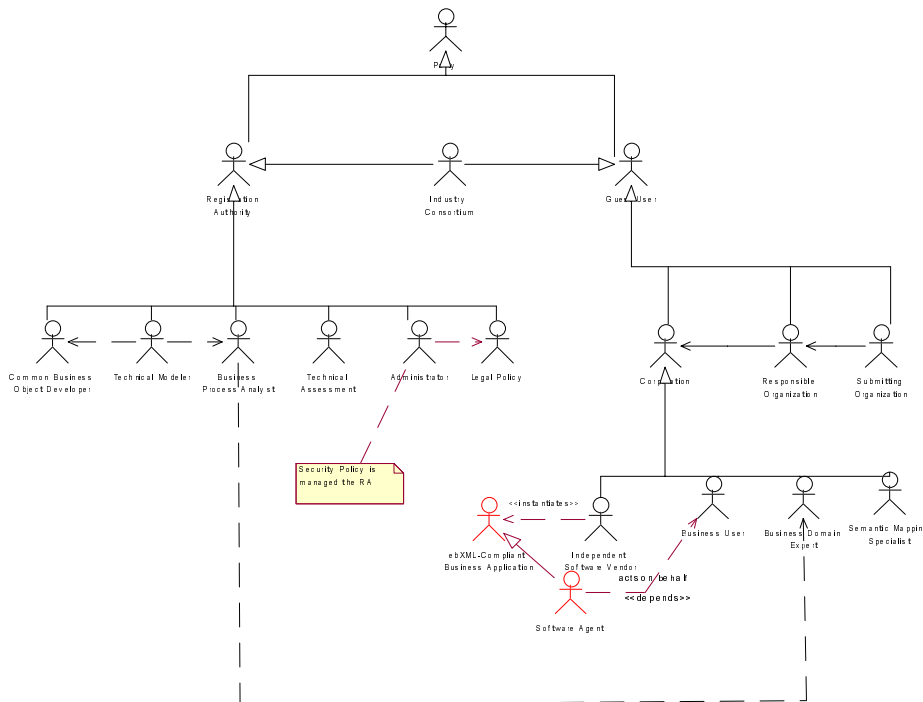
155 Many of the terms and definitions can be found in:

- 156
- 157 • ISO 11179
 - 158 • OMG UML Specification
 - 159 • TMWG Glossary
 - 160 • OMG XMI Specification

161
 162 They are not repeated here.

163 **2 Actor Relationships**

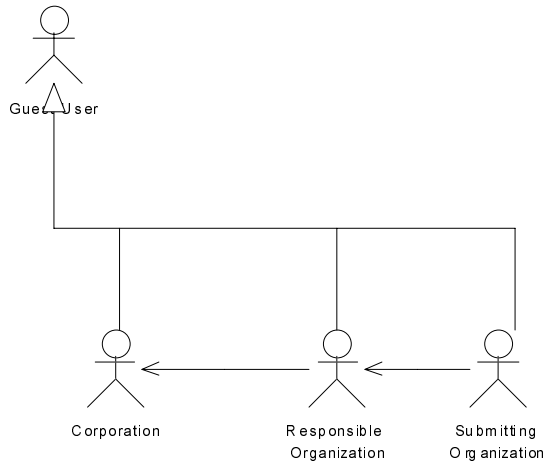
164 The complete Actor Relationship class diagram depicts the hierarchical relationships, the
 165 unidirectional associations, and the dependencies between the actors. The actor
 166 relationships are not intended to specify role base authorization. The registration authority
 167 shall define the security policy.
 168



169
 170 **Figure 2.**
 171 The main areas in this hierarchy are elaborated upon below.

172 **2.1 Guest User**

173 The Guest User is an actor that inherits from the superclass actor Party. The Guest User is
174 an organization or individual that acts on behalf of the organization.



175

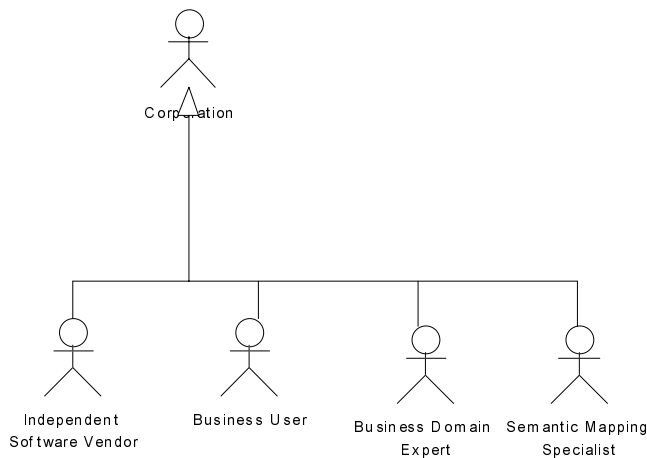
176 **Figure 3.**

177 Specialized actor classes of Guest User include the Corporation who is generally the end
178 user of the registry and repository, the Responsible Organization and the Submitting
179 Organization as defined by ISO 11179. Unidirectional associations are drawn between these
180 organizations; e.g., the Submitting Organization acts on behalf of the Responsible
181 Organization who in turns acts on behalf of the Corporation.

182 **2.2 Corporation**

183 The Corporation hierarchy does not have any other associations other than inheritance.
184 Shown are four types of actors, three of which could reside in a typical corporate business
185 environment; specifically, the Business Domain Expert, Business User and the Semantic
186 Mapping Specialist.

187



188

189 **Figure 4.**

190 The Business User could be from a large, medium or small corporate enterprise. The
191 Business Domain Expert is an actor that represents the corporate needs in the development

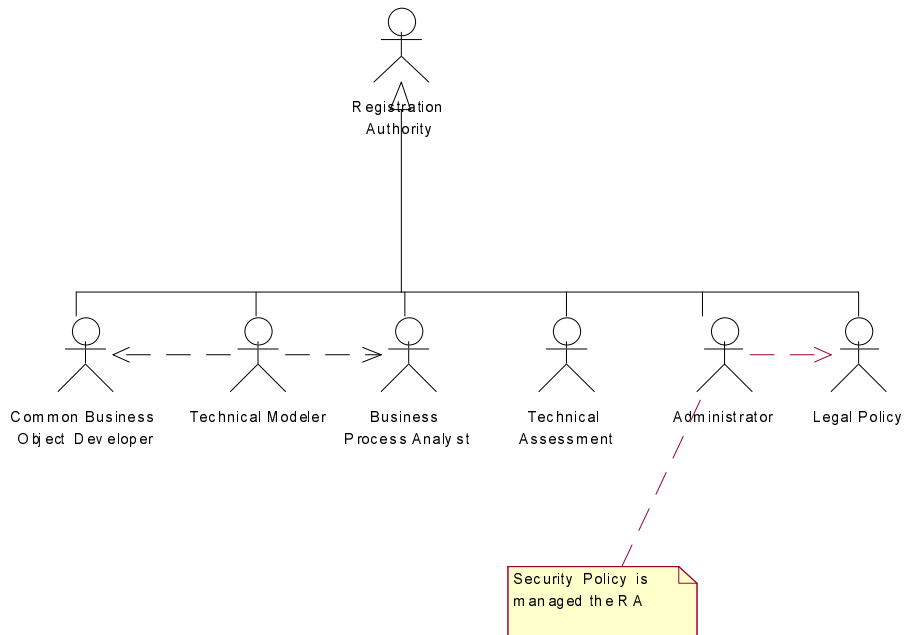
192 of B2B specifications. The Semantic Mapping Specialist is an actor that reviews available
193 B2B standards and specifications and maps them to the corporate internal business
194 processes and application metadata representations. The Semantic Mapping Specialist
195 traditionally supports EDI transactions or internal application-to-application (A2A) interfaces.
196

197 The Independent Software Vendor is a special type of Corporation that produces software
198 that meets the needs of the Business Domain Expert. The business needs of the Business
199 Domain Expert could imply that there is a unidirectional association between the Business
200 Domain Expert and the Independent Software Vendor, however those needs are
201 communicated through other channels, specifically through Technical Modelers (see
202 Registration Authority).
203

204 2.3 Registration Authority

205 The Registration Authority (RA) is shown with several actors that could actually be part of the
206 RA itself, subcontracted resources, or in some cases volunteers.
207

208



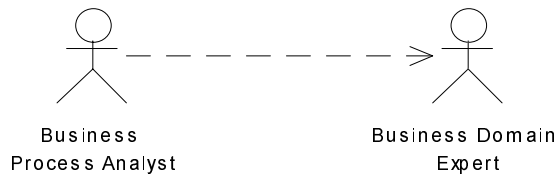
209

210 **Figure 5.**

211 Shown are six examples of roles within the Registration Authority. Three actors support the
212 knowledge extraction process to define the business and functional requirements of B2B
213 transactions: the Business Process Analyst, the Technical Modeler, and somewhat indirectly,
214 the Common Business Object Developer. These role players are found in the UN/CEFACT
215 Methodology for e-Business Process Modeling. The Business Process Analyst is a facilitator
216 with the skills to ask questions of the Business Domain Expert and is vertical domain “neutral”
217 in order to challenge the Business Domain Expert’s stereotypical views of processes, and to
218 identify business requirements including requirements that are not so obvious to the Business
219 Domain Expert but have been experienced in past by the Business Process Analyst.

220

221



222

223 **Figure 6.**

224 The Technical Modeler is familiar with the selected UML modeling tool and knows how to
 225 interact with the repository for model development and to search for existing patterns. The
 226 Technical Modeler is reliant on the Business Process Analyst skills in extracting knowledge
 227 from the Business Domain Expert. The Common Business Object Developer is skilled in
 228 reviewing UML models and detecting “patterns” in both business process models (activity
 229 diagrams and sequence diagrams) and data / design patterns (class diagrams). The
 230 Technical Modeler is also dependent on the Common Business Object Developer, who has
 231 the ability to purchase reusable common business objects for inclusion into the models in
 232 development.

233

234 The Technical Assessment actor is responsible for UML model consistency and clarity. One
 235 task of the Technical Assessment actor is to identify semantic overlap, where equivalent
 236 semantic information is represented in different ways. Technical Assessment works with the
 237 Common Business Object Developer to resolve such issues. The Technical Assessment
 238 analyst is responsible for reviewing an industry submission of a model, identifying the
 239 submission’s relevance, uncovering any semantic overlaps with other specifications in the
 240 repository, assigning model integration project plans, and issuing recommendations for the
 241 final approval of a proposed specification. Model integration is a step that harmonizes
 242 multiple submitted models that are indirectly inter-dependent to each other. This occurs
 243 primarily between vertical domains, in which each vertical domain names each semantic unit
 244 differently (ref: UN Layout Key).

245

246 The Administrator actor is responsible for the day-to-day operation of the ebXML Registry and
 247 / or Repository. This includes networking, server scalability, and usage statistics. The
 248 administrator has a thorough understanding of the overall *ebXML Architecture, ebXML*
 249 *Transport, Packaging, and Routing specifications*, and the APIs to the Registry and
 250 Repository as defined in *ebXML Registry and Repository Part 4: Design* (to track the number
 251 of API requests).

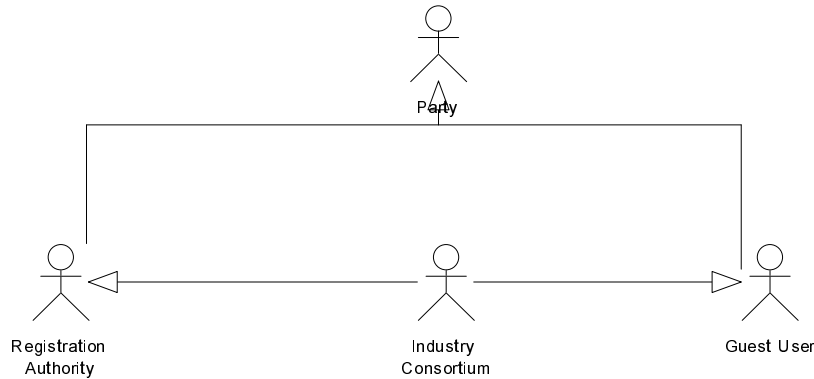
252

253 The legal/policy actor is responsible for the business arrangements, legal policies, and
 254 operational policies of the registry and repository.

255 **2.4 Industry Consortium**

256 The Industry Consortium can be a Registration Authority as well as a Guest User therefore
 257 multiple inheritance is shown. In reality, the inheritance to Registration Authority has optional
 258 cardinality, as the Industry Consortium may not choose to host its own ebXML-compliant
 259 Registry and Repository.

260

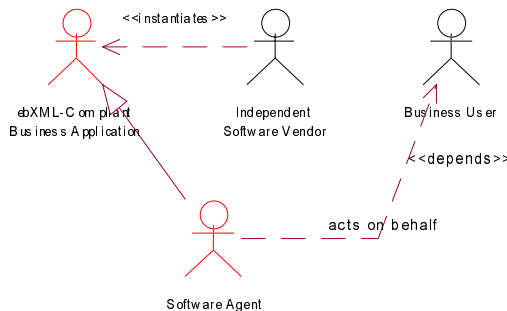


261
262
263

264 **Figure 7.**

265 **2.5 ebXML Business Application**

266 An ebXML Business Application is an application that knows how to communicate with the
267 ebXML-compliant registry. An independent software vendor may develop such an application
268 or it may be developed internally by a corporate development team using the ebXML
269 specifications for a given problem domain. This is shown by the “**instantiates**” relationship
270 between the Independent Software Vendor and the ebXML-compliant Business Application.



271
272

Figure 8.

273 A specialized actor class of the ebXML-compliant Business Application is the Software Agent.
274 This is shown by an inheritance relationship to the ebXML-compliant Business Application
275 actor. Software agents are based on artificial intelligence technology and are emerging in the
276 industry at a rapid pace. A software agent acts on behalf of the Business User actor who
277 configures the agent by specifying a unique profile that classifies a problem domain. This is
278 noted by the “dependency” association between the Software Agent and the Business User.
279 The unique behavior of the software agent is that it typically runs in an autonomous mode
280 without initiation by the Business User, and presents to the Business User information that
281 conforms to the profile. Software agents could monitor B2B transactions for anomalies,
282 detect trends, find new trading partners, and configure B2B integration systems.

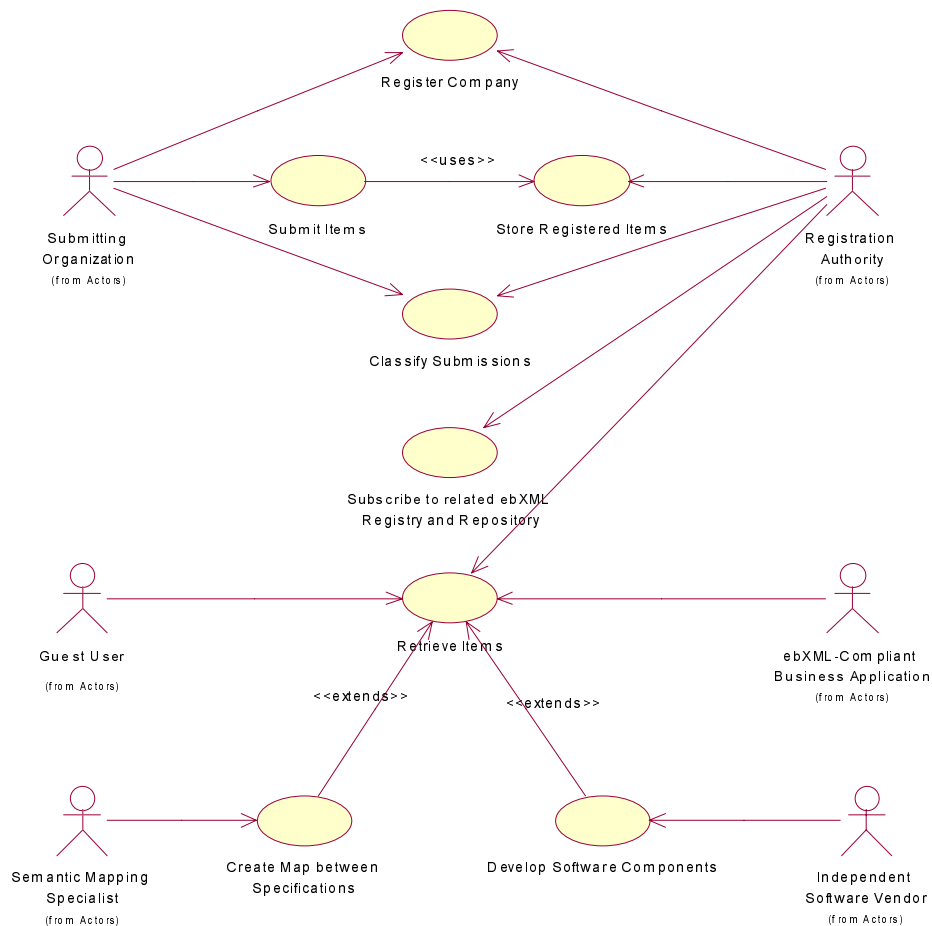
283
284
285
286

Software agents will be able to interact with an ebXML-compliant registry, and request specifications for e-Business Processes and any referenced data interchange specifications in the e-Business Process.

287 **3 Domain Use Case**

288 The overall scope of the registry and repository is the submission, classification, and storage
289 of specifications and proposals, while allowing the creation of new specifications. The
290 Domain Use Case provides an overall, high-level context of the business problem that doesn't
291 delve into how the system performs the tasks on a functional basis. It provides a boundary to
292 the scope of the functionality, and provides traceability to the actors involved as defined in the
293 Actor Relationships hierarchical diagram.
294

295 All actors are human actors with the exception of the ebXML Business Application. (It does
296 not appear as a different UML icon as suggested by other modeling conventions.) It is
297 important to refer back to the Actor Relationship diagram to understand the various
298 associations between the actors in the context of their roles. This approach enables a simple
299 Domain Use Case diagram as shown below.
300



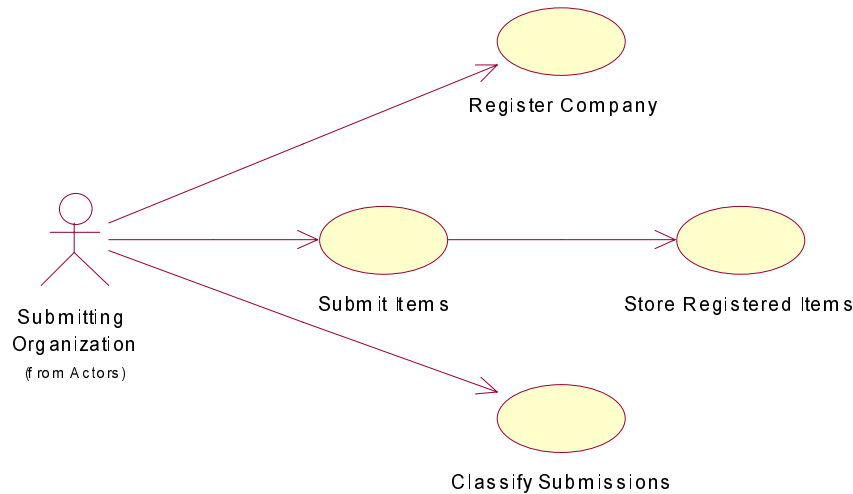
301
302 **Figure 9.**
303 Detailed discussion of each actor's viewpoint follows.

304 **3.1 Submitting Organization**

305 The Submitting Organization (SO) is responsible for the submission of items that are to be
306 included into an ebXML Repository and referenced in an ebXML Registry. The SO must first
307 register itself and be approved to submit items as noted by the Register Company use case.
308 After the SO has been approved, appropriate access rights are granted that allow the SO to

309 maintain its submissions and their metadata based on the different status codes specified by
310 *ebXML Registry and Repository Part 3: Analysis*.

311
312 The Submit Items use case provides the ability to submit a package. The Submit Items use
313 case always “uses” the Store Registered Items use case to enter the submission into the
314 repository. A record is created in the ebXML Registry (hence the term “register”), and a
315 submission object is created in the repository as a container of the submission, which may be
316 a collection of various documents, models, DTDs and other related information. Items within
317 the collection may also be registered in the registry, at the discretion of the SO. The RA may
318 at a later date create another collection, and relate that collection to the SO collection.
319



320

321 **Figure 10**

322 The SO can attempt to classify its submission, however, it is only a recommendation for the
323 classification as the Registration Authority may change or add classification information after
324 researching the submission.

325

326 3.2 Guest User

327 A Guest User is only granted read access. The Retrieve Items use case is abstract and
328 includes the ability to directly retrieve data from the Repository or to search the ebXML
329 Registry metadata in many ways, such as through unique identification, classification
330 schemes, browsing, or complex search mechanisms. In the latter instance, a request may be
331 sent from the Registry to the Repository to obtain a copy of the requested item.
332



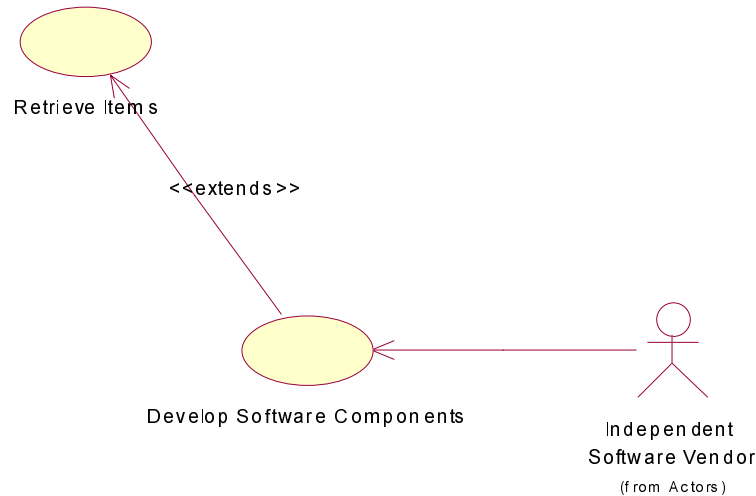
333

334 **Figure 11.**

335 There is an implied trusted security relationship between the Registry and Repository that
336 should be transparent to the Guest User. The Guest User may need to authenticate to the
337 Registry.

338

339 **3.3 Independent Software Vendor (ISV)**
 340 The Independent Software Vendor (ISV) actor uses the ebXML Registry and Repository to
 341 retrieve B2B specifications, in order to Develop Software Components. The Develop
 342 Software Components use case is extended by the Retrieve Items use case since at some
 343 point the specifications must be downloaded or viewed on-line. It is not required at all times
 344 (optional) throughout the software development life cycle, therefore the <<extends>>
 345 stereotype is used versus the <<uses>> stereotype.



346
 347 **Figure 12.**

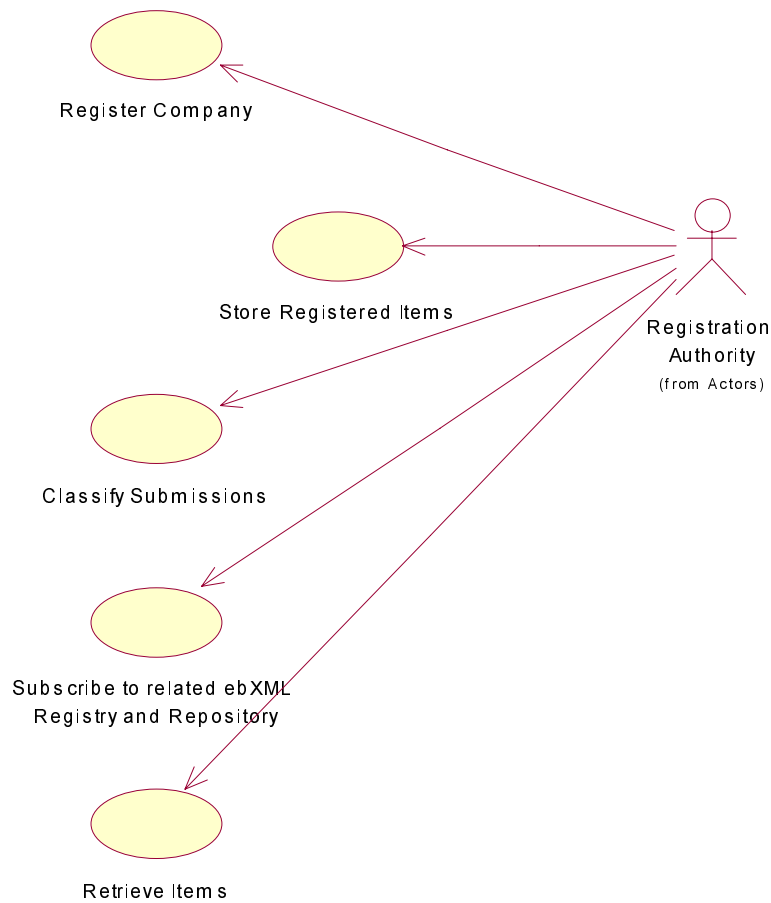
348 Since ISV is a special type of Corporation, the software components that are constructed,
 349 tested and certified to be compliant to the retrieved specification(s), the ISV can, through its
 350 association to the SO, register and submit the software component for download by the
 351 Business User.

352
 353 In addition, since the ISV is a special type of Guest User, the ISV may browse the registry
 354 and repository in a read-only mode to determine whether there are any good specifications
 355 that may be of interest. This allows the ISV to determine if the Develop Software
 356 Components use case should be entered.

357
 358 The work product from the ISV could become a revenue source for both the ISV as well as
 359 the RA to cover administration costs.

360 **3.4 Registration Authority (RA)**

361 The Registration Authority (RA) is the owner of the ebXML registry and an ebXML Repository.
 362 The RA authorizes the SO's registration request that was initiated in the Register Company
 363 use case. This may include a complete background check of the SO in question.



364

365 **Figure 13.**

366 The Store Items use case allows the RA the ability to grant public access to any items that the
 367 SO may have submitted. These items are in a queue until this grant occurs. As the trust
 368 between the SO and RA improves over time, the RA may certify the SO; i.e., allow the SO to
 369 Store Items without prior review by the RA. Factors that may go into certification criteria
 370 include the number of submissions, accuracy of the SO's classifications, and ability for an ISV
 371 to produce software components based on the SO's specification quality.
 372

373 **3.5 ebXML-compliant Business Application (ebAppl)**

374 The ebXML-compliant Business Application imposes run-time requirements on the ebXML
 375 Registry and Repository, causing response time to become a key factor in the Retrieve Items
 376 use case. If the ebAppl is smart (software agent), the search and retrieval of items can be
 377 more complex, and collections of items could be returned. Therefore, collections of items are
 378 in scope for Retrieve Items use case.
 379



380

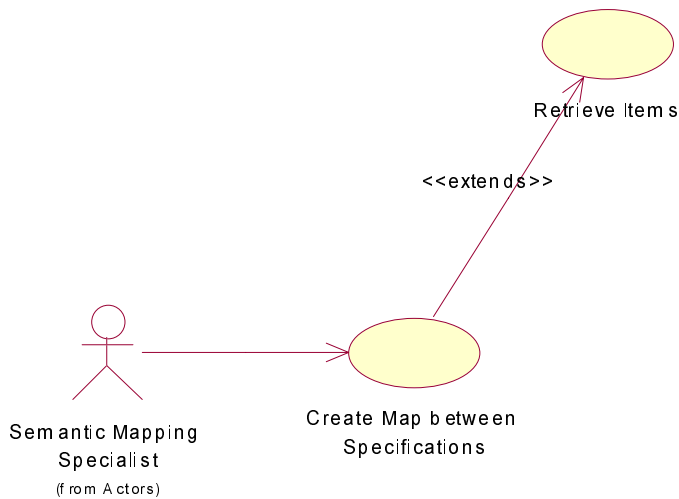
381

382 **Figure 14.**

383 An example of an ebXML-compliant Business Application interacting with an ebXML Registry
384 is to discover new trading partners that participate in a given marketplace, or provide certain
385 products and services. This type of data is supported by the ebXML Business Process
386 Metamodel, which is similar to eCo's Type Registry specifications. Conformance to eCo
387 Types is out of scope for this specification, and is referenced only to suggest that the eCo's
388 Type Registry can be implemented through an ebXML Registry.

389 **3.6 Semantic Mapping Specialist**

390 The usage of the ebXML Registry and Repository by the Semantic Mapping Specialist actor is
391 very similar in scope to that of the ISV, except that the creation of maps includes the retrieval
392 of two or more specifications. The interaction is the same in that the Create Map between
393 Specifications use case is extended by the Retrieve Items use case. Specifications can be
394 downloaded and mappings can be created off-line.



395

396 **Figure 15.**

397 The type of mappings could be "bridges" beyond two or more specifications, or integration
398 mappings to commercial software in which the vendor does not provide an integration
399 interface. Again, the Semantic Mapping Specialist is a special type of 1) Guest User for
400 read-only search and retrieval usage, and 2) Corporation which can submit its work through
401 its association to the SO.

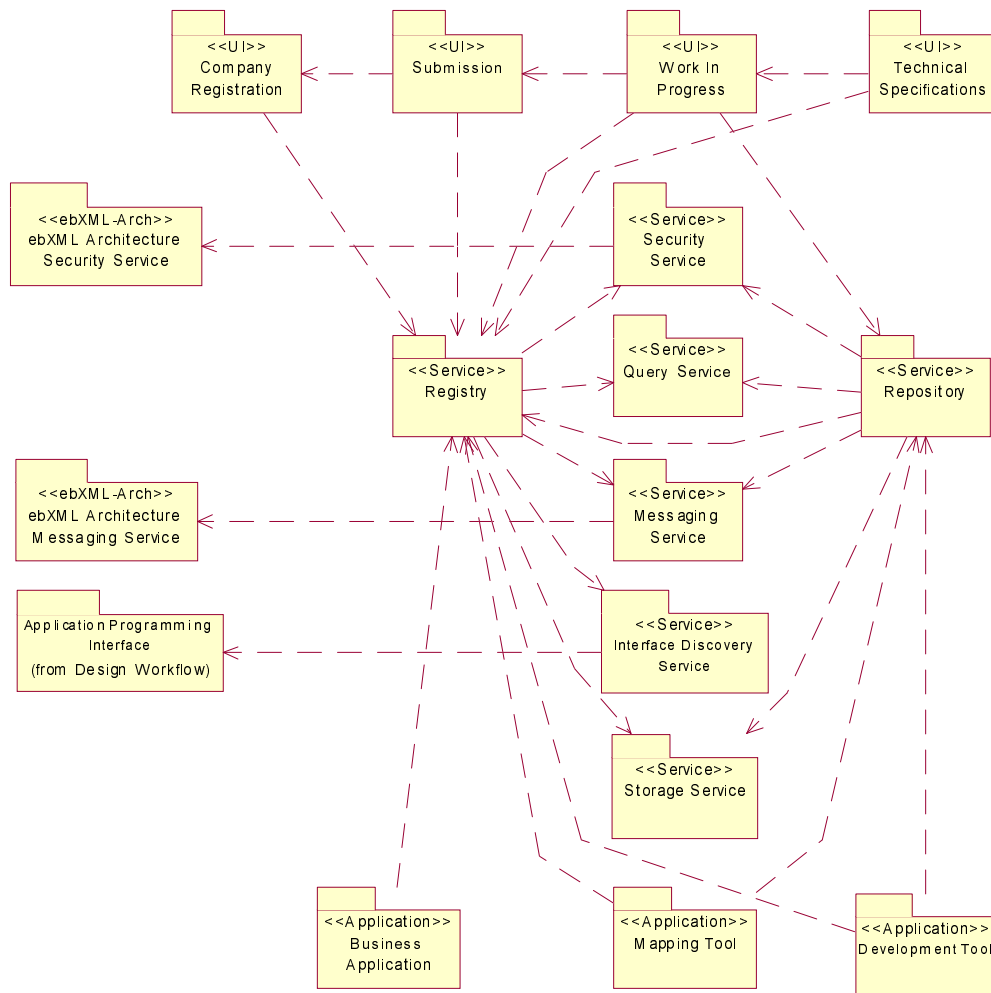
402

403 The work product from the Semantic Mapping Specialist could become a revenue source for
404 both the Semantic Mapping Specialist and the RA to cover administration costs.

405 **4 ebXML Registry and Repository Architecture (Domain Package Diagram)**

406 The following Domain Package Diagram details the various "subsystems" which carry out the
407 responsibilities for the domain use cases. A specific use case may be carried out by a
408 subsystem or an interaction between subsystems. The partitioning of the responsibilities for
409 each use cases into packages or subsystems provides for an architecture that is intended to
410 scale with size and increased volume of usage over time.

411



412

413 **Figure 16.**

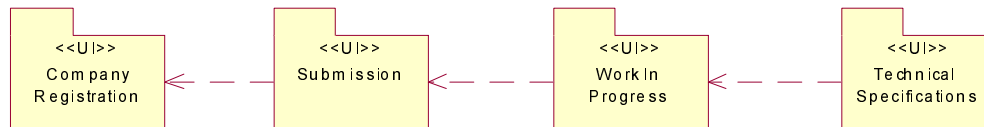
414 It is IMPORTANT to note that there are optional <<Service>> packages **within** the Registry
 415 and Repository <<Service>> packages that cover additional capabilities that may be
 416 performed by the Repository. Each of these services will have its own API. These services
 417 are noted as optional or mandatory, and reflect whether a repository can be implemented by a
 418 file-based system, which is the minimum condition for implementation.

419

420 This document discusses the high level statement of the scope of each package and the
 421 interrelationships. The document *ebXML Registry and Repository Part 2: e-Business*
 422 *Requirements* provides more detail. Specifically, each package will have one use case
 423 diagram, and each use case will have a detailed activity diagram that describes the workflow.
 424

425 4.1 Interrelationships between <<UI>>

426 The <<UI>> stereotype provides the scope for the User Interfaces and the workflow required
 427 for companies and individuals to interact with the registry and retrieve information from the
 428 repository. There are four main areas that comprise the registry and repository as shown in
 429 figure 17.



430
431

432 **Figure 17.**

433 This shows how to submit an item or package of items to a repository, allow people to be
434 aware that it has been submitted, and if desired, allow the submission to progress into an
435 ebXML-compliant specification. The Registration package is traceable to the Register
436 Company use case. It will provide the use case diagram and detailed workflows for each use
437 case in the diagram.

438

439 After an organization is registered with the RA and becomes an SO, it can submit items to the
440 Registry and store them in the Repository. The Submission package is traceable to the
441 Submit Items, Store Items, Classify Submissions, and Retrieve Items use cases. Unique
442 classification schemes may be needed for each type of submission, since some submissions
443 may have different important characteristics or attributes to describe them. Classifications will
444 be detailed in *ebXML Registry and Repository Part 3: Analysis*, which will define the class
445 diagrams for the registry.

446

447 If the SO intends for the submission to become an ebXML Specification to cover a business
448 process, the submission may need to be enhanced. The Work In Progress package
449 describes the workflow steps to become a Technical Specification, including library control
450 functions such as checkin/checkout and versioning. Types of workflow steps may include
451 enhancement of a submitted item to ensure that it meets the requirements of the ebXML
452 Business Process Metamodel, harmonization with existing common business objects (Core
453 Components) residing in the repository, and integration with other submitted items that may
454 have overlapping semantics. Overlaps may occur for different vertical domains that have
455 dependencies to each other. If a submission is not intended to be an ebXML Specification,
456 but perhaps only a corporate specification, it still may reside in the repository but classified
457 accordingly. In that case, there is no work to be performed, and the submission would pass
458 rapidly through the Work In Progress package and flow into the Technical Specifications
459 package.

460

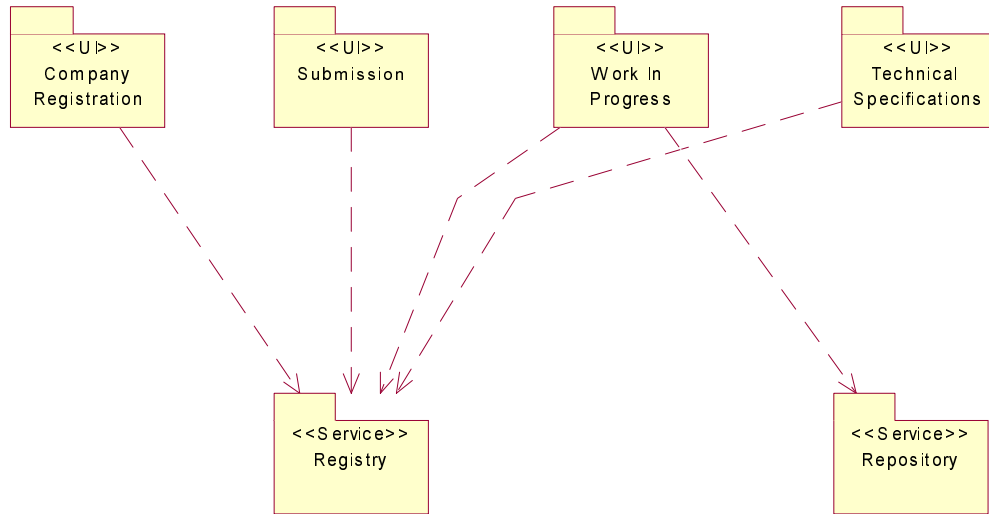
461 The Technical Specifications package allows the submission to become a specification that
462 can be viewed by **the public**. Work in progress is not available to the public unless in alpha
463 or beta form, and classified as such. The Technical Specifications package provides the
464 ability to browse the registry based on various classification schemes and to create simple
465 and complex searches to find specifications.

466

467 *4.2 Interrelationships between <<UI>> and <<Services>>*

468 In order for the <<UI>> packages to complete each step of the workflow, each package must
469 interact with the Registry package. The Work in Progress package may interface directly with
470 the Repository package to add modified copies when a library control service is provided with
471 the repository. Since the registry maintains state information about the original submitted
472 item, the original copy will always be retained.

473



474
475
476
477
478
479

Figure 18.

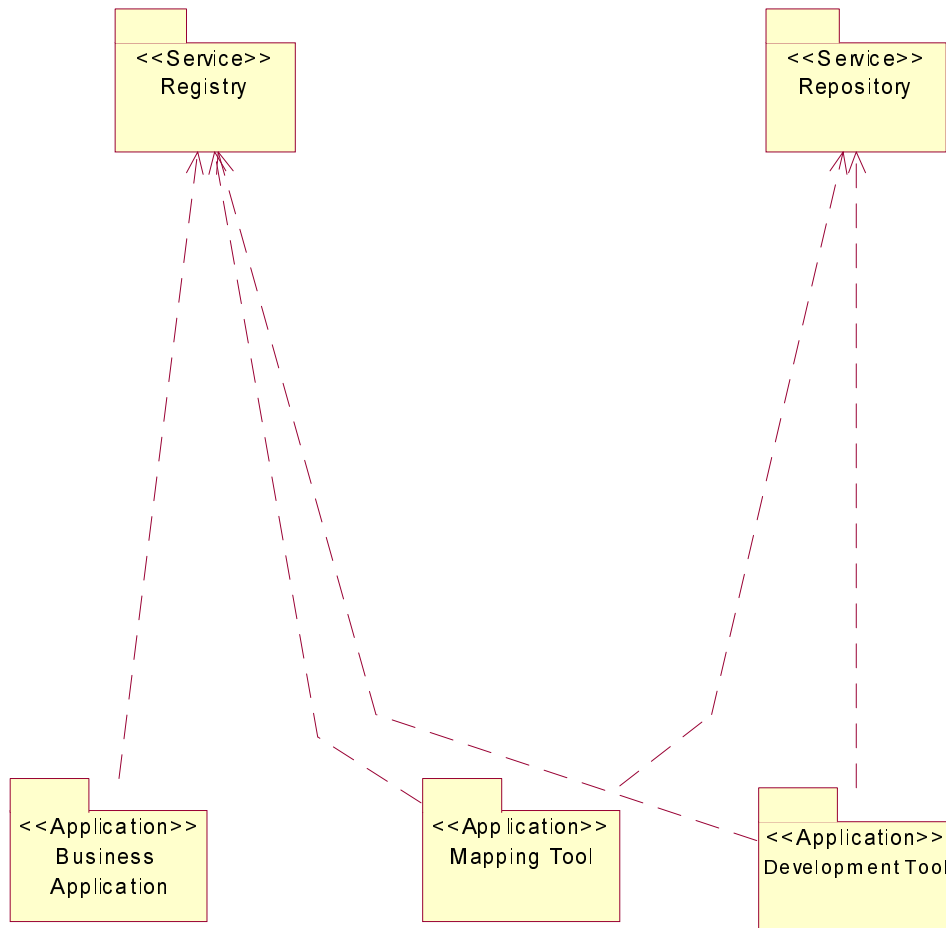
All interactions via the <<UI>> are done through the Registry, and not through the Repository. This allows a more scalable architecture, especially when some of the Registry contents are cached in memory.

480
481
482
483
484
485
486
487
488
489
490
491

4.3 Interrelationships between <<Applications>> and <<Services>>

The Registry and Repository also provide the ability for various development and business applications to interact with them.

As in typical software development life cycle, application development tools interact directly with the Repository. The two <<Application>> packages that are shown are the Mapping Tool, and the Development Tool. While they are both development tools *per se*, the Mapping Tools involves two or more specifications that need to be mapped at any one time. The Development Tool is a typical Integrated Development Environment including commercial Java, C++ and Smalltalk environments (VisualAge, Visual C++, Visual Café are examples). These IDEs require various features of the repository including check-out/check-in services, versioning services, transformation services for code generation as well as other features.



492

493 **Figure 19.**

494 The Business Application <<Application>> can only interface directly with the Registry, and
 495 has no need what for the repository services including check-out/check-in. This is traceable
 496 to the Retrieve Item use case in which the ebXML-compliant Business Application interacts
 497 with that use case.

498

499 **4.4 Interrelationships between <<ebXML-Arch>> and <<Services>>**

500 The Security and the Message services are related to the overall ebXML Architecture for
 501 security services and message services in conformance with the ebXML Transport, Package,
 502 and Routing specification. Thus the authentication, certificate, and encryption techniques are
 503 common across all ebXML-compliant Business Applications. Since the ebXML Registry and
 504 Repository is also a type of ebXML-compliant Business Application, intercommunication
 505 between the registry and repository is done using the ebXML TRP messaging service.

506



507

508

509
510

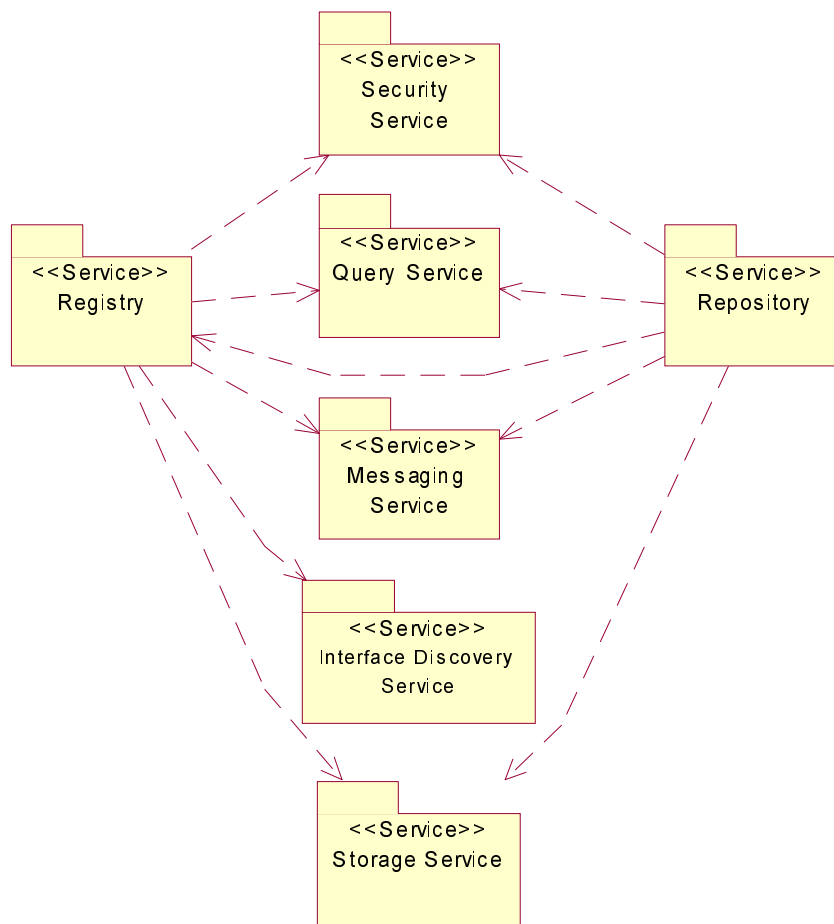


511 **Figure 20.**

512 **4.5 Interrelationships between <<Services>>**

513 As previously stated, the Registry and Repository both depend on a common security and
514 messaging service, as formerly illustrated.

515
516 For the simple case of retrieval of a registered item by reference to a unique identifier over
517 HTTP it is desirable to support "THHTTP" as specified by RFC-2483 and revised by the OASIS
518 registry and repository Technical Committee as well as TRP.



519

520 **Figure 21**

521 The dependencies between the Registry and Repository show a number of different features.
522 First, the Registry's dependency on the Repository shows a publish and subscribe
523 mechanism, in which the Registry receives metadata from a Repository particularly when a
524 specification reaches a state in which the information needs to become public, or a new
525 version is issued. The Registry may subscribe to more than one Repository besides its
526 Primary Repository. This dependency is traceable to the Subscribe to Related Repository
527 use case.

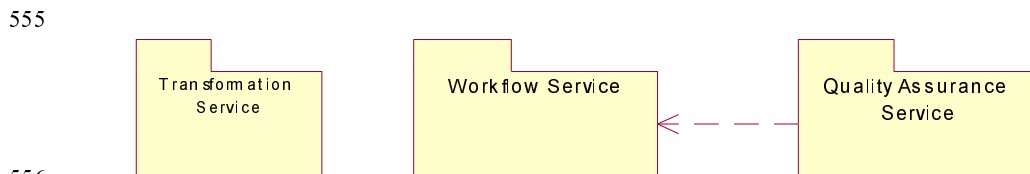
528
 529 A Primary Repository is an ebXML-compliant Repository maintained by the RA to which the
 530 Registry has write authorization. This is shown by the Repository's dependency on the
 531 Registry to receive its content and establish a URI link to the content. This URI link must be
 532 maintained by the Registry.
 533
 534 The dependencies on the query service package show that metadata in the registry can be
 535 searched to find information regarding contents of the repository. This metadata describes the
 536 contents in the repository including its URI for the ability to retrieve a copy of the item. In
 537 addition, based on the RA's policies or types of information stored, the repository content
 538 could be searched. The types of searches our defined in *ebXML Registry and Repository*
 539 *Part 3:Analysis*.
 540
 541 The dependencies on the storage service shows that registered items in the repository must
 542 be stored. In addition, the classification schemes and metadata are stored in the registry.
 543 The physical storage mechanisms are not specified, but could be as simple as a file system.
 544



545
 546 Figure 22

547 The Interface Discovery service package describe a mandatory service that must be
 548 implemented by the RA. The RA chooses which optional services they will provide. Interface
 549 Discovery service is the mechanism that advertises which interfaces are available. Based on
 550 the security policies set up by the RA, these interfaces may be restricted.
 551
 552 The application programming interface will be defined in *ebXML Registry and Repository Part*
 553 *4:Design*.

554 **4.6 Optional Services within the <<Service>> Registry package**



556
 557 Figure 23.

558 **4.6.1 <<Service>> Transformation Service**

559 The Transformation Service is used to transform objects into another form. (e.g., IDEF-1X to
 560 XMI, XMI to XML Schema). *This is going to be defined by ebXML Registry and Repository*
 561 *Part 2:e-Business Requirements*.

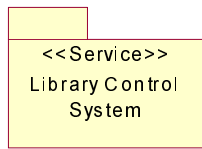
562 **4.6.2 <<Service>> Workflow Service**

563 The Workflow Service carries out the workflows that are defined by *ebXML Registry and*
 564 *Repository Part 2:e-Business Requirements*.

565 **4.6.3 <<Service>> Quality Assurance Service**

566 The Quality Assurance Service is used to validate content based on its classification and is
 567 only invoked on certain workflow events; e.g., registration of ebXML business process model.
 568 *This is going to be defined by ebXML Registry and Repository Part 2:e-Business*
 569 *Requirements*.

570 4.7 Optional Services within the <<Service>> Repository package



571

572 Figure 24

573 4.7.1 <<Service>> Library Control Service

574 The library control system package is an optional service that supports development, e.g., the
575 work in progress package. *This is going to be defined by ebXML Registry and Repository*
576 *Part 2:e-Business Requirements.*

577 Appendix

- 578 • [OASIS Registry and Repository Technical Committee](#)
- 579 • [UN/CEFACT UML Profile and Methodology](#)
- 580 • [OMG Meta Object Facility \(MOF\)](#)
- 581 • [OMG Unified Modeling Language](#) version 1.3 includes UML, OCL and XMI (located
582 on Rational's web site)
- 583 • [ISO 11179](#)
- 584

585

586

587

588