

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

Methodology for describing Core Components
Analysis and Proposal
(Draft-Rev.04)

By Work Group 7 of Core Component Project Team
Hisanao Sugamata
Martin Bryan

Contents

53	
54	
55	
56	1. Introduction
57	
58	2. Business entity analysis
59	2.1 Example <Party>
60	2.2 Requirements for defining the 'Party' entity within
61	UN/EDIFACT
62	
63	3. Common Business Entity concept
64	3.1 Representation Class candidates
65	3.2 Core Component Meta model
66	
67	4. Considerations
68	4.1 Reusability
69	4.2 Syntactically neutral
70	4.3 Basic Semantic Register
71	4.4 Naming the component
72	4.5 Segmentation of Core Components
73	
74	5. Describe the Core Component
75	
76	6. Sample description
77	
78	7. Instructions for capturing ebXML Core Component definitions
79	7.1 Defining a Pattern
80	7.2 Defining an Entity
81	7.3 Defining a Data Representation
82	7.4 Defining a Data Format
83	7.5 Defining a Code Set
84	
85	Appendix : Core Component Templates

86

87 **1. Introduction**

88

89 The first step for achieving to get specifications of the ebXML core components will be analysis of
90 the existing data models currently used in e-business. The concept model for describing the business
91 entities, which were nominated as the candidates of the core components at the 2nd project team
92 meeting of ebXML in Orlando, will be proposed in the paper. This paper will also offer the
93 templates for describing the results of analysis on the business entities.

94

95 The purposes of the paper includes,

96

97 1. To show the way to OO-edi methodology through the reverse engineering for the
98 current EDI messages for business experts, who have been engaged in designing
99 legacy EDI messages; and

100

101 2. To show how to analyze business entities currently being used in EDI in order to design
102 the object classes for object modeling experts, who have been engaged in designing the
103 software in the manner of Object Oriented Approach (OOA).

102

103 It is expected that business entities, which are selected and analyzed by the business experts of the
104 Core Component Working Group, can be used in the business process models. The entities can also
105 be stored in the repository(s) aligning with the ebXML standards.

106

107 The analysis and proposal prepared in this paper, respecting the ebXML requirements specification,
108 is intended to be,

109

110 1. Syntactically neutral,

111

112 2. Conforming to ISO11179 and

113

114 3. Aligning with the Unified Modeling Methodology.

115

116 4. A mechanization for developing Core Components

117

118 **Revision note :**

119

120 There have been several discussions around the first draft of this paper.

121

122 One of the disputing subjects was how to describe the analysis pattern for the business entities.

123

124 You can find the analysis and the proposal for describing the pattern in the section 4.5.

125

126 In the 3rd revision of the draft paper, Representation Class candidates are introduced in the section
127 3.1.

128

129 In addition to the previous version of the paper, Mr. Martin Bryan proposed the XML based
130 templates instead of the paper-based form for mechanizing the development of Core Components.

131

132 In the appendix file attached you can find the usable templates in XML based form.

133

134 In the 4th revision of the draft paper, the XML based templates are amended through some testing.

135

136 Also the meta model for Core Component is added in the section 3.2.

125 **2. Business entity analysis**

126

127 At the 2nd meeting of the core component team in Orlando, typical business entities were on the table
128 for discussion. One of of the business entities discussed was 'Party'.

129

130 **2.1 Example <Party>**

131 The business entity 'Party' is observed in several patterns or within contexts. The 'Party' is defined
132 within segment groups in the UN/EDIFACT messages.

133 In the UN/EDIFACT messages of ORDERS and INVOIC, the pattern of the business entity 'Party'
134 has rich attributes which is specially used in purchasing applications.

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

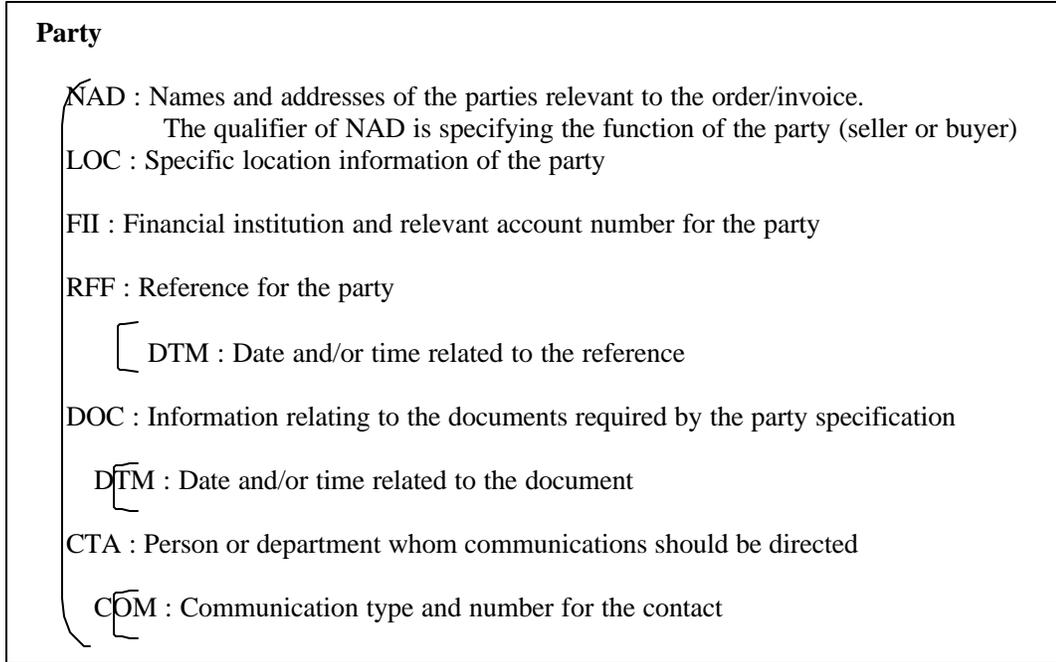
152

153

154

155

156



157

Fig.1 Party example – 1

158

159 On the other hand, the 'Party' entity in other message types have a simpler pattern. The simple
160 patterns can be seen in the messages, such as CUSDEC (Customs declaration message), DELFOR
161 (Delivery schedule message), BANSTA (Banking status message), BUSCRD (Business credit report
162 message) and IPPOAD (Insurance policy administration message). The typical pattern of the simple
163 form of 'Party' is as follows.

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

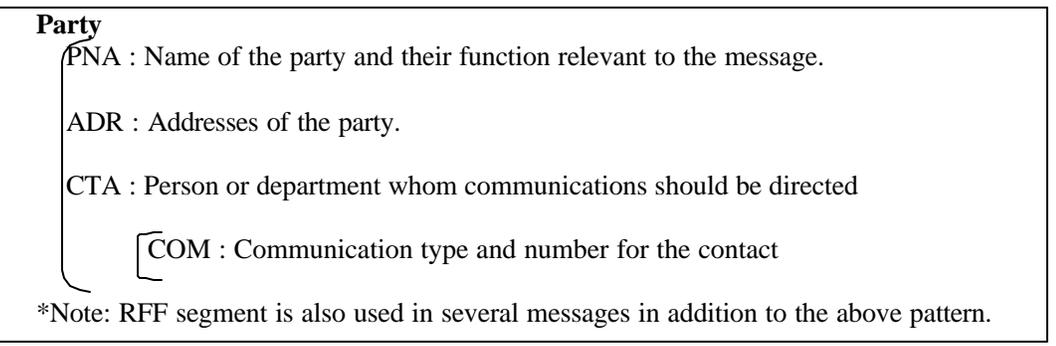


Fig.2 Party example – 2

The above examples of the pattern for 'Party' are representing the organizations involved in the

179 relevant business. Sometimes the one side of the business party may be an individual person. The
180 single person case can be seen in the application of the medical industry, the life insurance industry
181 or the labor market industry. The following sample shows the pattern in the message MEDREQ
182 (Medical service request message)

Party
PNA : Name of the party or professionals and their function relevant to the message.
ADR : Addresses of the healthcare party.
COM : Communication number for healthcare party.
FTX : Unstructured telecommunication numbers of the party (ex. Email)
RFF : Alternative identification number assigned to the party
SEQ : Allocated sequence number to the party.
LAN : The language used by the healthcare party.
SPR : The medical specialty of a healthcare organisation.
QUA : The professional medical qualification.
EMP : The position or rank of a healthcare professional.

195
196 Fig.3 Party example – 3

197
198 In the above example, the party can be the organization or the individual professional person.
199 When the entity 'Party' is used for the individual, another entity 'Person' is recommended. Using the
200 entity 'Person' avoids the complex pattern of the entity.

201
202 We should consider which pattern is efficient for being used in the e-business. There may be several
203 measurements to decide which is better for ebXML standard.

204
205 Two measurements shall be considered as the first priority. The measurements are (1) 'Reusability in
206 general applications' and (2) 'Usability in the specific application'. We should carefully model the
207 business entity patterns considering the tradeoff between two of them. (see Section.4)

208
209 In above samples of the business entity 'Party', we can choose the second one (Fig.2) as one of the
210 common business entities. We can choose it because it is used in various kinds of applications.

211 212 **2.2 Requirement for defining the 'Party' entity within UN/EDIFACT**

213 214 (1) Abstraction

215 The first segment of the 'Party' entity has a qualifier, which is specifying the function of the
216 party.

217 The first data element 3055 specifies the function of the party. (see Fig.4)

218 The party without the qualification has no meaning in the real business. The party can be 'the
219 seller', 'the buyer' or the other party who has the specific function in the relevant business. In
220 another words, the entity 'Party' is an abstraction for many roles in the business. We can call the
221 'Party' as a super class and the party who has the specific function in the business as a sub-class
222 in the object-oriented world. In the XML world the party would have meaning based upon the
223 context of where the party element was included in the hierarchical model.

224
225 <Note 2> The discussion on how to describe the relations between classes can be found in the
226 section 4.5.

227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279

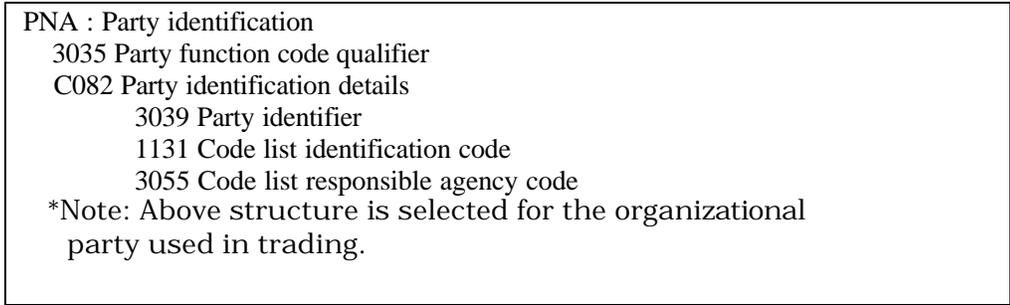


Fig.4 PNA segment structure

(2) Relation

The entity 'Party' is composed with the segments PNA(Party identification), ADR(Address), CTA(Contact information), COM(Communication contact) in UN/EDIFACT messages. PNA (Party identification) is used only in the entity 'Party', but others can be used in other relations of other entities. For example, ADR (Address) may be used for specifying the delivery address. Therefore, it is clever to distinguish the entity and the relation of the entities (the pattern). The entities and the patterns are the candidates for the core components of ebXML standards.

(3)Attributes

In the Fig.5, you can see several attributes for the entity 'Party'.

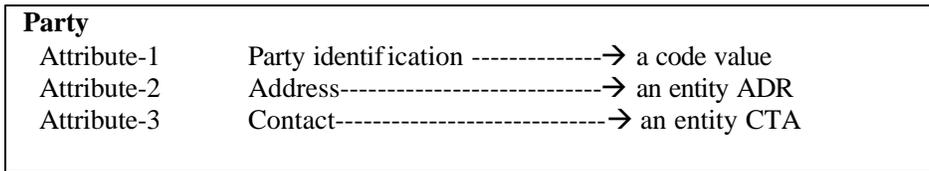


Fig.5 Attributes of Party

The first attribute 'Party identification' has a value that identify the party. The second and third attributes shall be specified through the other entities. The address of the party is specified by the attribute of the entity ADR. The telephone number or E-mail address for contacting the Party is specified through the entity CTA and COM.

(4) Representation

When the attributes get their values, the characteristics for them shall be defined. The identification of Party may be coded form, the address may be specified in Postal form, the telephone number may be numeric and the E-mail address may be character string. Any values of attribute shall have their certain representations.

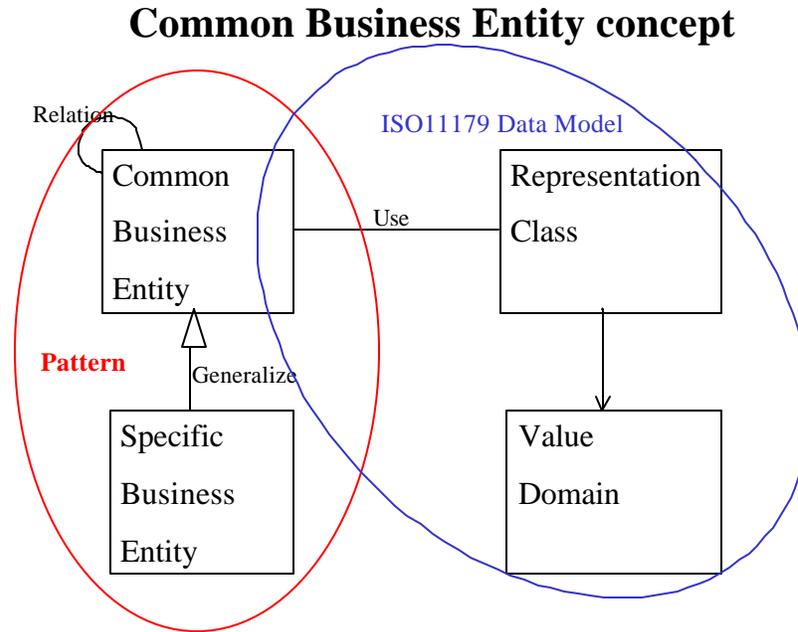
(5) Value

The first data element (3039) of the composite data element C082 in Fig.4 shall have the value 'Party identifier'. The value of 'Party identifier' shall be specified in a code list that is defined by the following two data elements. There can be many code lists. Even the same value of the data element 3039 has deferent meaning in the deferent code list. These code lists are called value domains. When the data element has the number representation, the value can be any arithmetical value. All the arithmetical values are one of the value domains. When the data element has the calendar date, the value can be specified in the Gregorian calendar dates. All the calendar dates are another value domain.

280
281
282

3. Common Business Entity concept

Through the analysis of the business entities and the patterns in the previous section, the concept model for Common Business Entity can be described as



283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302

follows.

Fig.6 Common Business Entity concept

- (1) Common Business Entity is a concrete class or an abstraction class generalized from one or more Specific Business Entities.
- (2) A concrete class of Common Business Entity or a Specific Business Entity has one or more instances.
- (3) Common Business Entity may have any kind of relations to other Common Business Entities. The series of relations related to a certain business behaviour is called Pattern.
- (4) Common Business Entity shall have attributes those are related to other Common Business Entities or shall use Representation classes.
- (5) Representation classes have properties. The properties for representation classes have a value domain, data type and, if necessary, a unit of measure or a character set.
- (6) The concept (lexical meaning) and the format (syntactical expression) are the two schemas of a data type.
- (7) The value domain is defined in the scope of certain concept (lexical meaning) with the certain format (syntactical expression).

303 The following items are kinds of Core Components.

- 304 (1) Analysis Pattern
- 305 (2) Common Business Entity
- 306 (3) Specific Business Entity
- 307 (4) Representation class
- 308 (5) Value domain

309

310 Analysis patterns may be used in the business process.

311 Analysis patterns, Common Business Entities and Representation classes shall be registered in a ebXML compliant repository.

312
313 The highly reusable Specific Business Entities shall be registered in a ebXML compliant repository. Other ones may be defined by each specific application for each specific industry.

314
315
316 The values in the highly reusable value domains shall be registered in a ebXML compliant repository. The application unique value domains, including code sets, may be defined by each specific application for each specific industry.

317
318
319

3.1 Representation Class candidates

320

321 The following lists are the candidates for Representation Class.

322
323 **amount** A number of monetary units. It is normally associated with a type of currency.

324
325 **code** A character string that represents a member of a set of values.

326
327 **description** A series of sentences describing a person, object, place, event or concept.

328
329 **identifier** A character string used to identify and distinguish uniquely, one instance of a value within an identification scheme.

330
331
332 **name** A word or phrase that constitutes the distinctive designation of a person, object, place, event or concept. What the person, object, place, event or concept is known by or called.

333
334
335
336 **number** An arithmetical expression representing a particular value. Note: This may often be used to imply sequence or a member of a series.

337
338
339 **percent** A rate expressed in hundredths between values that have the same unit of measure.

340
341 **quantity** A number of non-monetary units. It is normally associated with a unit of measurement.

342
343 **rate** A quantity or amount measured with respect to another measured quantity or amount.

344
345 **date or time** A date and/or time as measured in the time dimension.

346 **age** A length of time that a person or thing has existed.

347 **Indicator** An attribute indicating a condition such as on/off, true/fouls, yes/no, 0/1.

348 **measure** A standard unit used to express size, amount or degree.(*)

349 (*). Another idea is to use 'height', 'width', 'length', 'degree' or 'size' instead of 'measure'.

350

3.2 Core Component Meta model

351

352 The figure 7 shows the meta model for Core Component.

353
354

355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407

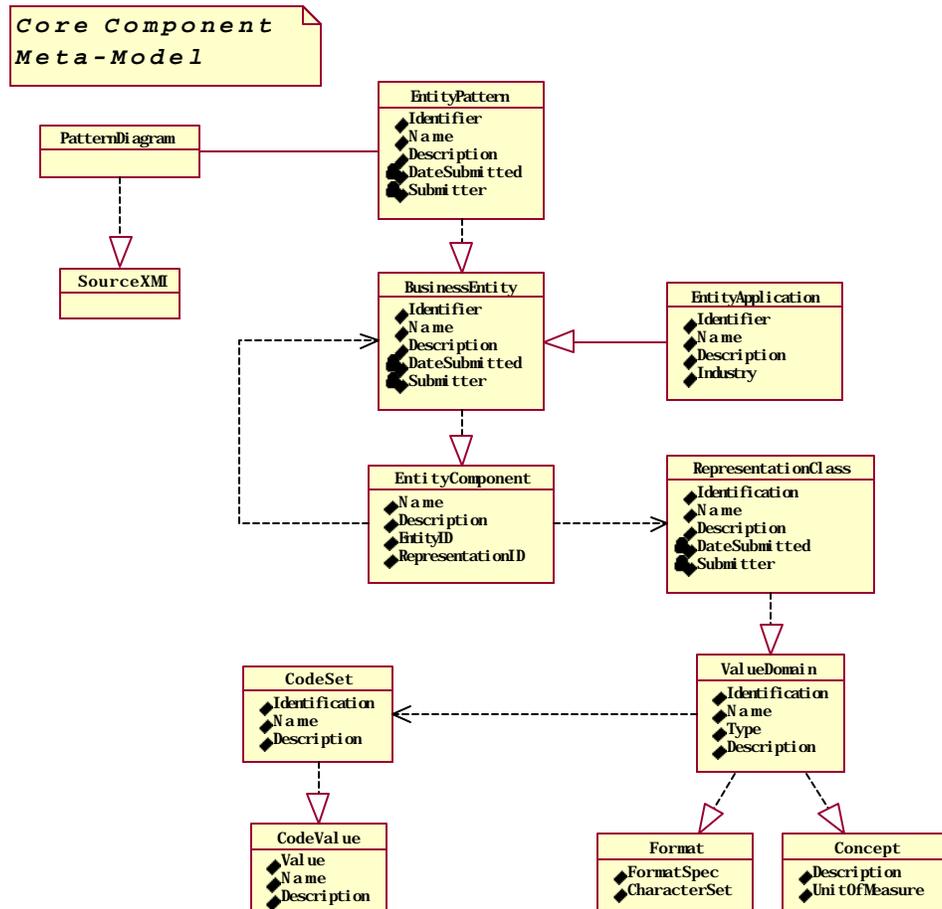


Fig 7 Core Component Meta model

4. Considerations

4.1 Reusability

The reusability of the core component is the key concern for efficiency of the e-business application software development. Reusing the standardized core components can push forward the interoperability between the e-business applications. On the other hand, having too much attachment to the widelevel usability in any application, you may recognize that only atomic level objects can be standardized, such as Numeric, Monetary amount, Gregorian calendar or Percentage. These atomic level objects are highly reusable. However, any other structured objects or object patterns have to be developed when you implement any e-business application. It makes it very difficult to implement the interoperability of applications.

When highly constructed objects are agreed upon for interchange, you can minimize your development efforts for implementation of the e-business application. The turnkey application packages or the fixed format EDI messages are easy to implement if all the parties involved are using the same platform (hardware) and the same application package (software). But it is impossible for a single uniformed application to be installed in all the enterprises in various industries. Therefore, the high level constructed components can be used only in

408 certain applications in certain industries, or can be used only between certain
409 trading partners.
410 Because of the above considerations, we shall select and standardize the proper
411 level of components.

412

413 **4.2 Syntactically neutral**

414 It is true that no model can be described without any syntax. This may be one of FDTs (Formal
415 Descriptive Technique), natural languages or graphical charts. The meaning of a syntactically neutral
416 model is a model free from an implementation level syntax, such as EDIFACT, XML or JAVA. In
417 other words, the model described in a syntactically neutral manner can be implemented in the
418 computer system using any proper implementation level syntax. Also the syntactically neutral model
419 described by the certain FDT for modeling can be mapped to another model using another FDT.

420 UML (Unified Modeling Language) may be one of the FDTs used for the syntactical neutral
421 modeling.

422

423 **4.3 Basic Semantic Register (BSR)**

424 According to the definition of TC154-BSR project, BSC (BSR Semantic Component) is a generic
425 term comprising the components of BSR semantic units. In the context of the BSR there are two
426 types of BSR semantic component, representation class and concept. The concept of BSC is almost
427 same as Common Business Entity of Core Components, and the representation class of BSR is same
428 as Representation class of Core Components. When Common Business Entities and Representation
429 classes are selected and specified, the work done by TC154-BSR project may be referred.

430 BSU (BSR Semantic Unit) is concept unambiguously defined, independently of any particular
431 physical representation, and which is semantically complete. It is independent of the process or
432 application in which it is used. It is constructed using BSR semantic components. But BSU is a
433 completely deferent approach from Core Component analyzed in this paper. BSUs represent
434 attributes of Specific Business Entity. In the concept of Common Business Entity, attributes are
435 defined in Common Business Entity and Specific Business Entities inherit the attributes of Common
436 Business Entity.

437

438 **4.4 Naming the component**

439 There are three purposes of naming the component.

- 440 1. To identify the component uniquely in the certain domain if there are no identifiers other than
441 the name.
- 442 2. To be recognized easily by human.
- 443 3. To specify the domain structure.

444 The data element conforming ISO11179 shall have the unique identifier other than the name,
445 therefore, the name of the data element is the primary means of identification of objects and
446 concepts for humans. Otherwise, the name is the only identification for the component in UML. Also
447 the element name is the only identification for the component in XML. However, an element name
448 in XML may have different definitions depending upon the context of the element within the XML
449 structure.

450 We need some identification methodology for naming core components even when we start to
451 analyze Business Entities. At the analysis phase, the identifiers should be recognized easily by both
452 humans and computers.

453

454

455 **4.5 Segmentation of Core Components**

456 This consideration looks at how it might be possible to use some of the less commonly used features
457 of UML to create sets of core components that can be reused in multiple contexts.

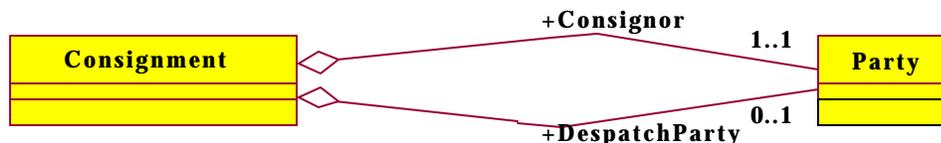
458 In the paper on *Transformation from EDIFACT to XML* Pharos group members from the EDIFACT
459 Transport group suggest that multiple associations should be used to identify the different “roles” a
460 particular set of data elements play. They state that:

461 “A role name defines a task or duty of a class in an association with another class.” The example

462 they use is the qualifier of the Party segment that indicates whether the party concerned is the Buyer,
 463 Seller, Consignor, Dispatch Party, etc.
 464 The Pharos document also contains a number of “rules” for the creation of UML models to represent
 465 business messages. Rule 5 states:

466
 467 **“Pharos Rule 5: Create multiple associations**

468
 469 Multiple associations are created for message classes that include subsections of the role list
 470 category.



473
 474
 475 Fig. 8 Example of multiple associations.
 476

477 Each of the multiple associations is given a multiplicity and a role name. The role names are
 478 found as data term names in subsections of the role list category.

479 The role name is used in the generated XML/DTD or XML Schema as a ‘group’ name for
 480 the attributes in the child message class structure. The UML notation uses a ‘+’ before the
 481 role name to indicate that the role name is ‘public’, but this symbol will be stripped in the
 482 generated XML/DTD or XML Schema.

483 The example multiple associations in the figure can be read as:

484 A Consignment has minimum 1 and maximum 1 Consignor

485 A Consignment has minimum 0 and maximum 1 DispatchParty”

486
 487 Whilst the use of associations to distinguish between the roles played by multiple occurrences of a
 488 class that can occur more than once within a document is highly commendable, it cannot be agreed
 489 upon with the group name assigned within the context. It is difficult to determine programmatically
 490 the context for a class within the XML DTD or Schema. A class with multiple roles should be
 491 represented by an abstract class in the UML model, with specific instances of the class being
 492 identified in the XML DTD or Schema by means of the name that associates the abstract class with
 493 the message.

494 The section 2.2 (1) takes a more traditional EDI approach to the purpose of these “qualifiers” of data
 495 element groups. It states:

496 The idea that Party is a super class suggested here might be better mapped to the concept of an
 497 abstract class, because there is no real sense of inheritance needed in this example

498 The comment on this concept was that “Assigning a ‘role’ qualifier to a party specification means
 499 that the information therein cannot be easily reused as part of an alternative role. If the role of the
 500 party is defined by its context, i.e. by its parentage/container within an XML tree, then reusability of
 501 Party-related information will be easier to manage. Unfortunately XML trees do not work in the
 502 same manner as traditional OO classes. We need to be able to identify things like
 503 Order\Buyer\PartyName and Order\Seller\PartyName rather than Order\Parties\Buyer and
 504 Order\Parties\Seller, which is what is implied if the party role defines a sub-class of Party. (Note the
 505 fact that the XML container needs to be Parties, not Party, if you use a class-based model, as the
 506 container is intended to contain information about all parties in current context.)”

507 In trying to reconcile the views, the sequence of containment can be expressed as:

508
 509 Order
 510 Parties
 511 Seller

512 Party Information
513 Party Identification
514
515

516 In discussions of the UK Data Harmonization Group at e-centre^{uk} it was pointed out that the Parties
517 and Party Information component of this sequence were really just containers that allowed the
518 correct management of data that forms part of the same abstract class. The other three components of
519 the “tree” represent the Business Process being undertaken (Order), the association of the abstract
520 class containing party information with the business process (Seller), and the core components that
521 identify the information to be interchanged for the completion of that part of the business process
522 (Party Identification ...).

523 The ‘real abstract class’ in this example is the Party Information abstract class. This may need to
524 take a number of different forms within different messages. For example, in the EB-Simpl model
525 most of the occurrences of data elements making up the Party Information are pre-exchanged, in
526 order that messages used by business processes only need to pass a key of the pre-exchanged
527 information (the Party Identification code). In other messages a wider range of the fields from the
528 Party Information set needs to be interchanged.
529

530 **5. Describe the Core Component**

531 Each business entity pattern can be described as the class diagram with the definitions of the
532 involved entities. But highly reusable representation classes shall be defined independent from any
533 business entities using them.
534

535 (1) Describing the business entity pattern

536 The set of the documents describing the business entity pattern is as follows.

- 537 - Entity pattern definition (One cover sheet)
 - 538 Entity patter ID
 - 539 Entity pattern name
 - 540 Entity pattern description
 - 541 Class diagram representing the entity pattern
- 542 - Entity definitions (One sheet for each related entity)
 - 543 Entity ID
 - 544 Entity name
 - 545 Entity description
 - 546 Attribute list
 - 547 Attribute ID
 - 548 Attribute name
 - 549 Attribute type (Designating entity or Using representation)
 - 550 Reference identification for designating entity or using representation
 - 551 Parameters for using representation class
 - 552 Attribute description
 - 553 Sub-class list
 - 554 Sub-class ID
 - 555 Sub-class name
 - 556 Sub-class function

558 (2) Describing representation class

559 Each representation class can be described in one sheet.

- 560 - Representation class definition
 - 561 Representation class ID
 - 562 Representation class name
 - 563 Representation class description
 - 564 Data type definitions
 - 565 Concept (lexical meanings)

566	Concept ID
567	Concept specification
568	Unit of Measure if needed
569	Format (syntactical expressions)
570	Format ID
571	Format specification
572	Character set if needed
573	Value domain list
574	Value domain ID
575	Value domain name
576	Value specification in the domain
577	
578	

579 6. Sample description

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

Pattern Definition	
Date: 24/Mar/2000	
Name: H. Sugamata	
Pattern ID : SAMPLE-P-01	
Pattern Name : Identification and contacting a party	
Description : To identify the party, who is involved in the business process, with their contacts.	

595 **Class diagram**

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

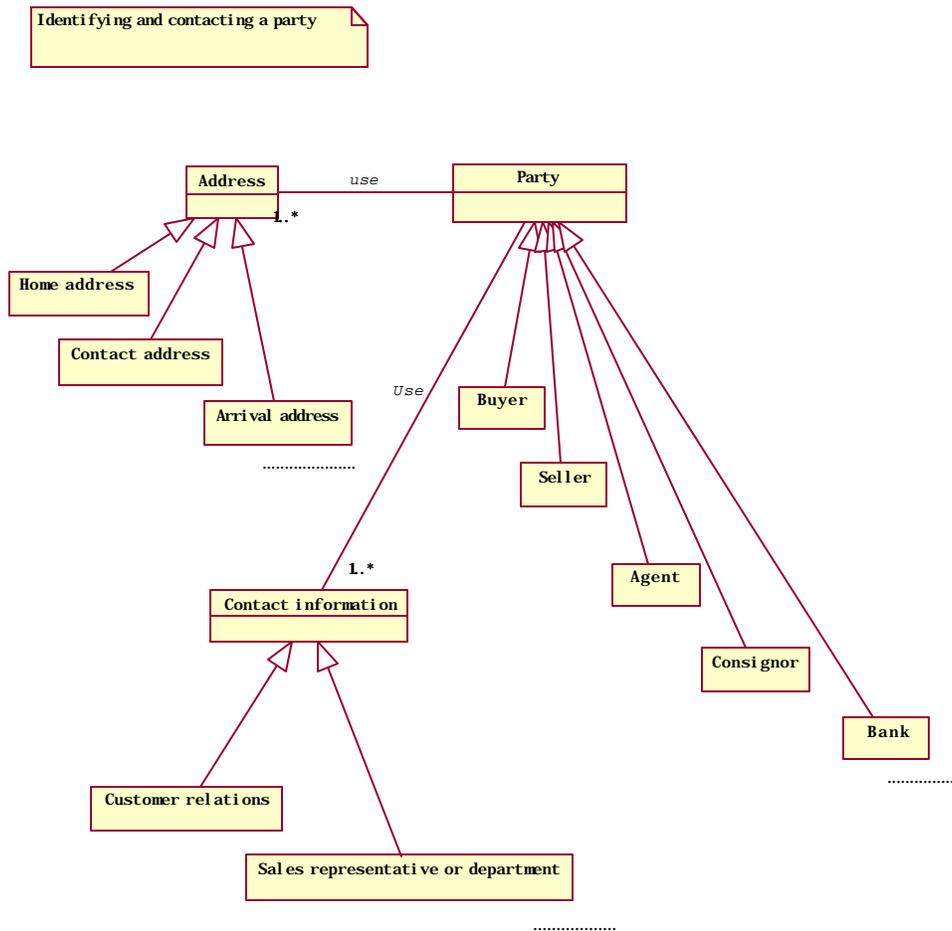
625

626

627

628

629



*Note: If there can be too many Sub-classes, Package may be used.

630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681

Business Entity Definition

Date: 24/Mar/2000
Name: H. Sugamata

Entity ID: SAMPLE-E-01
Entity Name: Party
Description: Unique framework of authority designating to act toward some purpose in the business.

Attribute List

Attribute name	Type	Ref.ID	Description
Party identifier (UN: 7402)	R	SAMPLE-R-01 (C1, F1, Vn)	Identification of the party.
Address	E	SAMPLE-E-02	Address of the party.
Contact information	E	SAMPLE-E-03	To identify a person or a department of the party, to whom communication should be directed.

*Note1: Type is R (using Representation class) or E (designating Business Entity).
 *Note2: 3 Parameters shall be specified with Representation class in Ref.ID field.
 Cn : Parameter of the Ref.ID for Concept
 Fn : Parameter of the Ref.ID for Format
 Vn : Parameter of the Ref.ID for Value domain

Sub-Classes

Sub-Class Name	Function
Buyer	Party to whom merchandise and/or service is sold.
Seller	Party selling merchandise to a buyer.
.....etc	
	The sub-classes of Party are defined in the code list of the data element 3055 (Party function code qualifier) in UN/EDIFACT directory. There are 498 functions for Party in D.00A.

*Notes: If there are too many sub-classes, you can designate the relevant code set.

682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734

Business Entity Definition

Date: 24/Mar/2000
Name: H. Sugamata

Entity ID: SAMPLE-E-02
Entity Name: Address
Description: To specify an address.

Attribute List

Attribute name	Type	Ref.ID	Description
Address type code (UN:3131)	R	SAMPLE-R-02 (C1, F1, V1)	Code specifying the type of an address.
Address status code (UN:3475)	R	SAMPLE-R-02 (C2, F1, V2)	Code specifying the status of an address.
Address component description (UN:3286)	R	SAMPLE-R-03 (C1, Fn, Vn)	Free form description of the component of an address.
City name (UN:3164)	R	SAMPLE-R-04 (C1, F1, V1)	Name of a city.
Postal identification Code (UN:3251)	R	SAMPLE-R-02 (C3, F2, V3)	Code specifying the postal zone or address.
Country name code (UN:3207)	R	SAMPLE-R-02 (C4, F3, V4)	Identification of the name of the country
Country sub-entity name (UN:3228)	R	SAMPLE-R-04 (C2, F1, V2)	Name of a country sub-entity.
Location name code (UN:3225)	R	SAMPLE-R-02 (C5, F4, V5)	Code specifying the name of the location which is defined in UNLOCODE.
*Note1: Type is R (using Representation class) or E (designating Business Entity).			
*Note2: 3 Parameters shall be specified with Representation class in Ref.ID field.			
Cn : Parameter of the Ref.ID for Concept			
Fn : Parameter of the Ref.ID for Format			
Vn : Parameter of the Ref.ID for Value domain			

Sub-Classes

Sub-Class Name	Function
Home address	The address is the home address.
Contact address	Address where contact may be made.
Arrival address	Address of arrival.
.....etc	The sub-classes of Address are defined in the code list of the data element 3299 (Address purpose code) in UN/EDIFACT directory. There are 7 functions for Address in D.00A.

*Notes: If there are too many sub-classes, you can designate the relevant code set.

735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787

Business Entity Definition

Date: 24/Mar/2000
Name: H. Sugamata

Entity ID: SAMPLE-E-03
Entity Name: Contact information
Description: To identify how to contact a person or a department to whom communication should be directed.

Attribute List

Attribute name	Type	Ref.ID	Description
Communication address (UN:3148)	R	SAMPLE-R-05 (C1, F1, Vn)	A communication address of a department or a person to whom communication should be directed.
Department or employee name code (UN:3413)	R	SAMPLE-R-02 (C6, F2, V6)	Code specifying the name of a department or employee.
Department or employee name (UN:3412)	R	SAMPLE-R-04 (C3, F1, V3)	Name of a department or employee .

*Note1: Type is R (using Representation class) or E (designating Business Entity).
 *Note2: 3 Parameters shall be specified with Representation class in Ref.ID field.
 Cn : Parameter of the Ref.ID for Concept
 Fn : Parameter of the Ref.ID for Format
 Vn : Parameter of the Ref.ID for Value domain

Sub-Classes

Sub-Class Name	Function
Customer relations	Individual responsible for customer relations.
Sales representative or department	The sales representative or department contact within an organization.
..... etc	
	The sub-classes of Contact are defined in the code list of the data element 3139 (Contact function code) in UN/EDIFACT directory. There are 95 functions for Contact in D.00A.

*Notes: If there are too many sub-classes, you can designate the relevant code set.

788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840

Representation Class Definition

Date: 29/Mar/2000

Name: Hisanao Sugamata

Representation ID :SAMPLE-R-01

Representation Name : Identifier

Description:

A character string used to identify and distinguish uniquely,
one instance of a value within an identification scheme.

Data type

Concept (lexical meanings)

Specification	Unit of Measure
1 Party identifier	
2	
3	

Format (syntactical expressions)

Specification	Character set
1 An..35	
2	
3	
* The formats of identifiers are defined in each schema.	

Value domain list

Domain name	Specification
1 Duns	Dun & Bradstreet Corporation assigned identifier
2 S.W.I.F.T.	S.W.I.F.T. assigned identifier
3etc	
*The identifier schemas are defined in the code list of data element 1131/3055 in UN/EDIFACT directory.	

841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892

Representation Class Definition

Date: 29/Mar/2000
Name: Hisanao Sugamata

Representation ID :SAMPLE-R-02
Representation Name : Code
Description A character string that represents a member of a set of values.

Data type

Concept (lexical meanings)

Specification	Unit of Measure
1 Address type code	
2 Address status code	
3 Postal identification code	
4 Country name code	
5 Location name code	
6 Department or employee name code	
7 Etc	

Format (syntactical expressions)

Specification	Character set
1 an..3	
2 an..17	
3 a2	
4 an..25	

Value domain list

Domain name	Specification
1 The code list for types of Address.	Refer DE3131 of UN/EDIFACT
2 The code list for status of address	Refer DE3475 of UN/EDIFACT
3 The code list for postal identification	The code lists are assigned by the country authority
4 The code list for countries name	Codes specified in ISO3166
5 The code list for locations	UNLOCODE specified UN/ECE recommendation 16
6 The code list for departments or employees	Code specified by organisation concerned

893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946

Representation Class Definition

Date: 29/Mar/2000
Name: Hisanao Sugamata

Representation ID :SAMPLE-R-03
Representation Name : Address component description
Description A description of the component of an address.

Data type

Concept (lexical meanings)

Specification	Unit of Measure
1 Address component description	
2	

Format (syntactical expressions)

Specification	Character set
1 Address format : Street name followed by number	
2 Address format: Number, road type, road name in this sequence	
3 Address format: Road type, road name, number in this sequence	
4 Address format: Post office box	
5 Address format: Unstructured address	
6 Address format: Street name followed by number, building, suite	
7 Address format: Rural route number	
8 Address format: Post office drawer number	
9 Address format: Building name followed by suite	
Note:1 Address formats are specified in the code list of the data element 3477 in UN/EDIFACT	

Value domain list

Domain name	Specification
1 Address component Description -1	Street name followed by number
2 Address component Description -2	Number, road type, road name in this sequence
3 Address component Description -3	Road type, road name, number in this sequence
4 Address component Description -4	Post office box
5 Address component Description -5	Post office box
6 Address component Description -6	Street name followed by number, building, suite
7 Address component Description -7	Rural route number
8 Address component Description -8	Post office drawer number
9 Address component Description -9	Building name followed by suite

947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

Representation Class Definition

Date: 29/Mar/2000
Name: Hisanao Sugamata

Representation ID :SAMPLE-R-04
Representation Name : Name
Description: A word or phrase that constitutes the distinctive designation of a person, object, place, event or concept. What the person, object, place, event or concept is known by or called.

Data type

Concept (lexical meanings)

Specification	Unit of Measure
1 City name	
2 Country sub-entity name	
3 Department or employee name	

Format (syntactical expressions)

Specification	Character set
1 an..35	
2	
3	

Value domain list

Domain name	Specification
1 City name	Name of a city in alphabetic characters
2 Country sub-entity name	Country sub-entity name in alphabetic characters
3 Department or employee name	Name of a department or employee in alphabetic characters

*Notes: If there are too many domains, you can designate the relevant code set.

1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051

Representation Class Definition

Date: 29/Mar/2000
Name: Hisanao Sugamata

Representation ID :SAMPLE-R-05
Representation Name : Communication contact address
Description: A communication contact address of a department or a person to whom communication should be directed.

Data type

Concept (lexical meanings)

Specification	Unit of Measure
1 Communication address	
2	
3	

Format (syntactical expressions)

Specification	Character set
1 an..512	
2	
3	

Value domain list

Domain name	Specification
1 International telephone	Telephone number, including country and/or city code as required, for voice or data transmission by telephone beyond the border of a country.
2 World Wide Web	Data exchange via the World Wide Web.
3 Electronic mail	Exchange of mail by electronic means.
.....	
Note:1 The value domain is defined in the data element 3155 in UN/EDIFACT. There are 34 value domains specified in D.00A.	

1052 **Instructions for capturing ebXML Core Component definitions**

1053

1054 The following instructions describe how Microsoft's Internet Explorer 5.0 (IE5) can
1055 be used to capture core component definitions as XML files.

1056

1057 WARNING: This process will not work on Netscape Explorer or on older versions of
1058 Internet Explorer as it relies on Microsoft specific extensions to HTML.

1059

1060 The UML metamodel for ebXML core components recognizes the following types of
1061 record:

1062

- 1063 - Patterns: UML models that define a set of related data entities, and the
1064 associations between them
- 1065 - Entities: UML model components that identify sets of related
1066 components, including data elements with a specific data
1067 representation
- 1068 - Date Representations: the set of data patterns or code lists that can be
1069 used to capture a particular component of an entity
- 1070 - Code Sets: A list of permitted values that can be used to complete a
1071 component, together with descriptions of their meaning
- 1072 - Data Formats: Details of constraints to be placed on the contents
1073 particular components of a pattern.

1074

1075 The HTML forms provided in this suite allow each of these data types to be
1076 recorded. To allow the relationships between forms to be clearly identified the
1077 basic UML model has been extended by adding elements that reference the ID of
1078 the next lowest level of component in the model. (This means that once a
1079 component, data representation, code list or data format has been defined once it
1080 need only be referenced in subsequent models.)

1081

1082 For each of the above record types there is an form with the appropriate name
1083 (pattern.htm, entity.htm, representation.htm, CodeSet.htm and DataFormat.htm).
1084 Each form contains buttons that allow the current contents to be submitted for
1085 storage, reset so that a new entry of the same type can be made or request a form
1086 for a record at a lower level in the metamodel.

1087

1088 The package has been designed to be run within a directory called ebxml within
1089 the My Documents section of your C drive. If wish to use another drive or root
1090 directory you should search each of the above files for any occurrences of the
1091 string "C:/My Documents/ebxml/" and replace this with the appropriate identifier
1092 for the directory you wish to use.

1093

1094 Within the directory that is used to contain the forms you will need to create five
1095 subdirectories prior to using the forms. These subdirectories should be labelled
1096 patterns, entities, representations, codesets and dataformats respectively. A
1097 further directory, called class, can also be defined to store class diagrams, etc.
1098 (Directories with these names may or may not have been created for you when you
1099 unzipped the source files, depending on your settings of your file unzipper.)

1100

1101 Because the forms submit their contents to the local file store rather than to a
1102 remote directory you will need to ensure that IE5 has been set up to permit this.
1103 In the Tools Menu select the Internet Options entry and then the Security tab. If
1104 your security level is High you will not be able to use these forms. If it is set to

1105 Medium you will be asked to confirm that each record may be written to disc. If it
1106 is Low you will be able to write the files without having to confirm each one, but
1107 must take care if using the Internet. (I find that the Medium level, which requires
1108 me to confirm before writing but still provides for a safe level of Internet access is
1109 an adequate compromise.)

1111 7.1 Defining a Pattern

1112
1113 The following figure shows the fields used to define a new pattern for use as a core
1114 component. The status line at the foot of the form will change as you move from
1115 field to field to provide you with information about the type of data to be entered in

The screenshot shows a web browser window titled "ebXML Data Representation Class - Microsoft Internet Explorer". The address bar shows "C:\My Documents\ebxml\pAttem.htm". The main content area is titled "ebXML Pattern Definition" and contains the following fields:

- Date of Submission: 2000-05-31
- Submitting Organization: UK-DHG
- Contact E-mail Address: mtbryan@diffuse.org
- Industry Sector: Retailing
- Business Process: Purchasing
- Pattern ID: Party-001
- Directory for Pattern storage: c:/My Documents/ebxml/patterns/
- Pattern Name: Party
- Pattern Description: Party involved in Simpl-eb purchase
- URL of Class Diagram: c:/My Documents/ebxml/class/Party-001.gif
- URL of XMI Representation of model: c:/My Documents/ebxml/class/Party-001.xmi
- ID to be assigned to root Component of Pattern: Party-001

At the bottom of the form, there is a status bar that reads "Assign Unique ID which can be used as filename of pattern".

1116 each field.

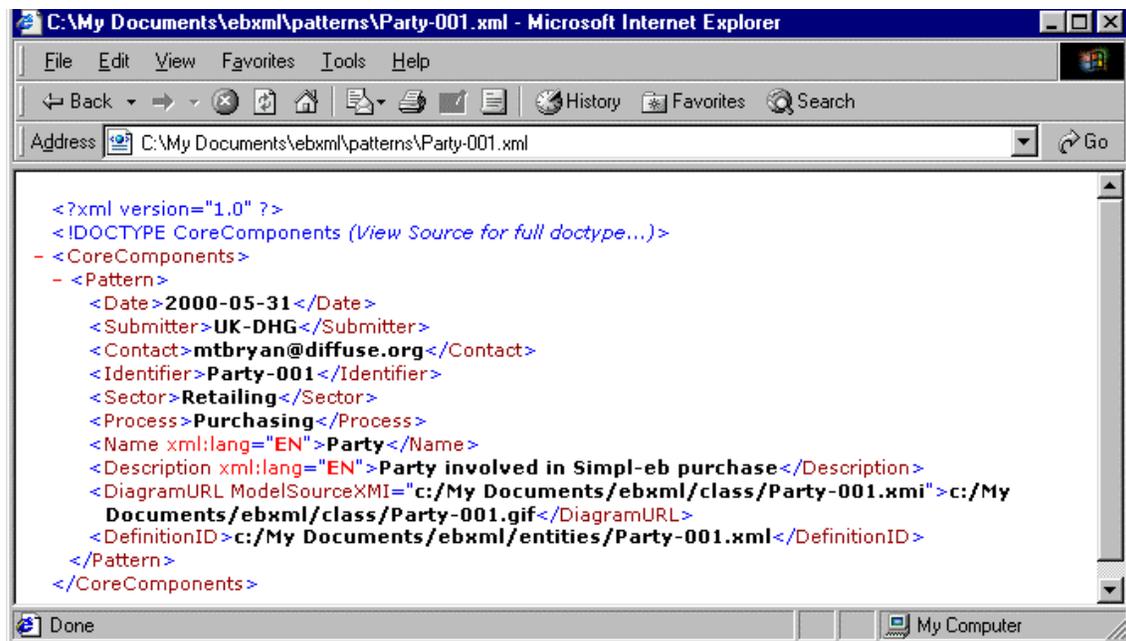
1117

1118 The fields provided are:

- 1119 - Date of Submission: enter an ISO 8601 conformant date using the
- 1120 format CCYY-MM-DD here
- 1121 - Submitting Organization: enter the name of the submitting
- 1122 organization here Contact E-mail Address: enter e-mail address to
- 1123 which questions can be submitted
- 1124 - Industry Sector: identify which communities the pattern is expected to
- 1125 be used by (using SIC codes where available)
- 1126 - Business Process: record business process(es) to which pattern
- 1127 applies (using SIC codes where available)
- 1128 - Pattern ID: Enter a unique identifier for the pattern. (This will form the
- 1129 file name of the pattern definition.)
- 1130 - Directory for pattern storage: Should indicate the path required to

- 1131 reach the required directory. (When the default directory set up is
 1132 being used this entry should not need to be updated from the pre-
 1133 assigned values.)
- 1134 - Pattern Name: Name to be used to identify the pattern. (May or may
 1135 not be the same as the Pattern ID.)
 - 1136 - Pattern Description: Enter description of role of pattern that allows it
 1137 to be distinguished from other patterns.
 - 1138 - URL of Class Diagram: Enter URL of file containing printable version
 1139 of class diagram. (For ease of use on the web we recommend this be a
 1140 GIF file. If this is stored in the class subdirectory within you ebxml
 1141 directory then all you need to do is to replace the ??? in the default
 1142 name displayed with the filename.)
 - 1143 - URL of XMI Representation of model: If an XMI representation of the
 1144 model is available for interchange enter the appropriate file reference
 1145 here: otherwise delete the default value.
 - 1146 - ID to be assigned to root Component of Pattern: Indicate which ID you
 1147 expect to assign to the Entity which will form the root of the pattern
 - 1148 - Directory that will be used for Entity storage: Should indicate the
 1149 path required to reach the required directory. (When the default
 1150 directory set up is being used this entry should not need to be
 1151 updated from the pre-assigned values.)

1152
 1153 When all relevant fields have been completed click on the Submit this Pattern
 1154 button. This will cause the inbuilt program to store an XML record pattern and
 1155 then display the contents of the file as shown below.



1156
 1157 Once you have ascertained that this file correctly records the details of your
 1158 pattern the window can be dismissed from the screen.

1159
 1160 The Pattern form ends with a button that allows you to "Create Entity Definition
 1161 for Root Component". Clicking on this button will call up the form needed to
 1162 record each of the entities defined within the pattern.

1163
1164
1165
1166
1167
1168
1169

7.2 Defining an Entity

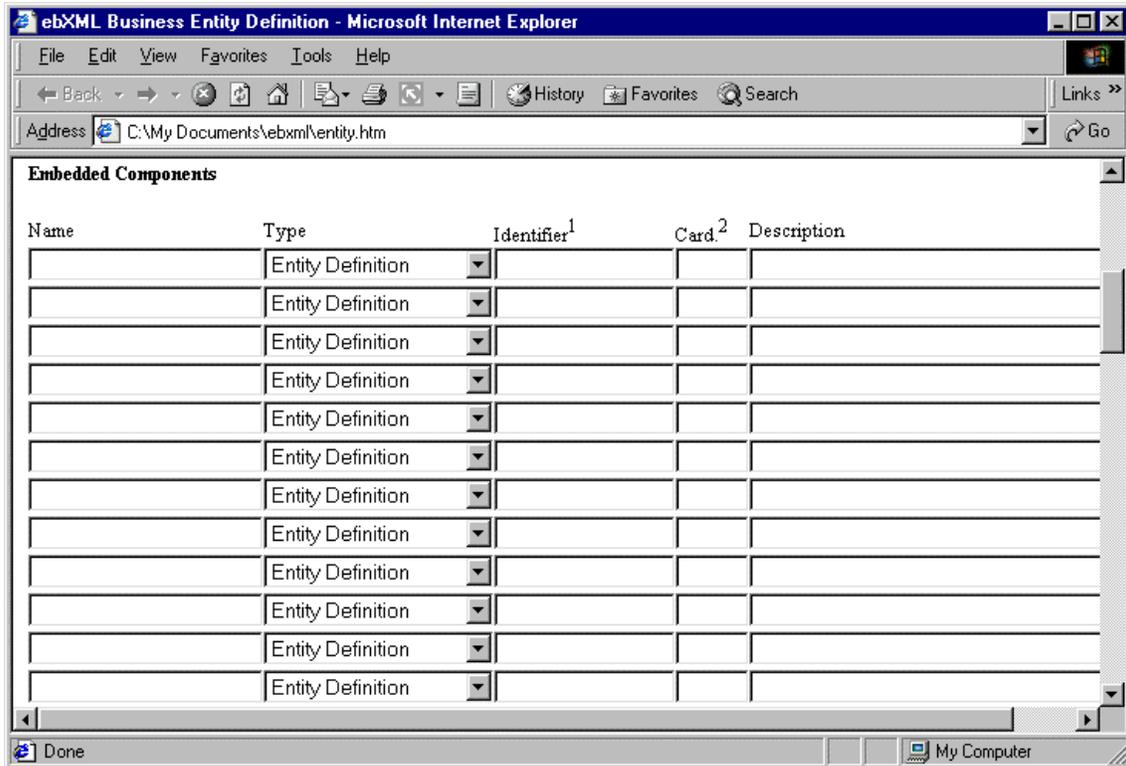
The following figures show the fields used to define a new entity for use within a core component. The status line at the foot of the form will change as you move from field to field to provide you with information about the type of data to be entered in each field.

The screenshot shows a web browser window titled "ebXML Business Entity Definition - Microsoft Internet Explorer". The address bar shows the URL "C:\My Documents\ebxml\entity.htm". The main content area displays the title "ebXML Business Entity Definition" in a large, bold font. Below the title, there are several input fields for defining an entity:

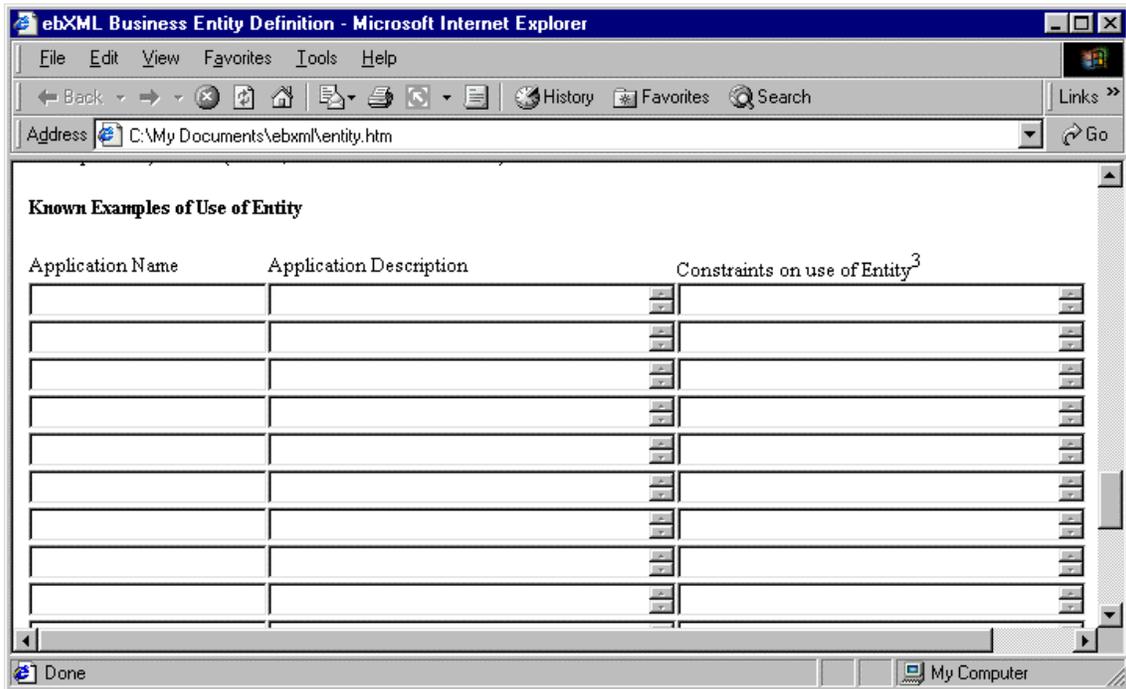
- Date of Submission:
- Submitting Organization:
- Contact E-mail Address:
- Industry Sector: Business Process:
- Entity ID: Directory for Entity storage:
- Entity Name:
- Entity Description:

The status bar at the bottom of the browser window shows "Done" and "My Computer".

1170
1171



1172



1173

1174 The fields provided in the first part of the form are:

- 1175 - Date of Submission: enter an ISO 8601 conformant date using the
- 1176 format CCYY-MM-DD here
- 1177 - Submitting Organization: enter the name of the submitting
- 1178 organization here
- 1179 - Contact E-mail Address: enter e-mail address to which questions can

- 1180 be submitted
- 1181 - Industry Sector: identify which communities the pattern is expected to
- 1182 be used by (using SIC codes where available)
- 1183 - Business Process: record business process(es) to which pattern
- 1184 applies (using SIC codes where available)
- 1185 - Entity ID: Enter a unique identifier for the entity. (This will form the
- 1186 file name of the entity definition.)
- 1187 - Directory for entity storage: Should indicate the path required to reach
- 1188 the required directory. (When the default directory set up is being
- 1189 used this entry should not need to be updated from the pre-assigned
- 1190 values.)
- 1191 - Entity Name: Name to be used to identify the entity. (May or may not
- 1192 be the same as the Entity ID.)
- 1193 - Entity Description: Enter description of role of entity that allows it to
- 1194 be distinguished from other core components.
- 1195

1196 Each line in the section headed Embedded Components has the following fields:

- 1197 - Name: Name of component in form suitable for use in XML DTD (i.e.
- 1198 starting with a letter and containing no spaces)
- 1199 - Type: Type of component (either Entity Definition if there are
- 1200 embedded components or Data Representation if this component is a
- 1201 root one designed to transfer data between systems)
- 1202 - Identifier: Indicate which ID you expect to assign to the
- 1203 Entity/Representation when you define it (this will create a cross
- 1204 reference to the definition you will create at a subsequent stage using
- 1205 either another copy of this form or the form for recording data
- 1206 representations.)
- 1207 - Card. If the embedded component is optional and/or repeatable the
- 1208 cardinality of this component with respect to its parent (i.e. how many
- 1209 times it can occur within the parent) as defined in the UML model
- 1210 should be recorded here, expressed as 0..1 (optional), 0.* (optional
- 1211 and repeatable), 1..* (required and repeatable) or m..n (m=min, n=max
- 1212 no of occurrences).
- 1213 - Description: Brief description of role of component within Entity.
- 1214

1215 If the entity is a pattern that is intended to be sub-classed or used as an abstract
 1216 class that is associated with concrete classes, examples of the intended use of the
 1217 entity can be recorded in the section headed Known Examples of Use of Entity.

1218 The following fields can be used for each example:

- 1219 - Application Name: Name by which sub-class or association is known
- 1220 - Application Description: Description of purpose of application
- 1221 - Constraints on use of Entity: Details of any constraints that apply to
- 1222 the use of the entity within this application (e.g. components that
- 1223 must or may not be used for this application of the pattern/entity.)
- 1224

1225 **Note:** At present there is no formal language for defining such constraints but in
 1226 future it is anticipated that text based descriptions entered initially will be
 1227 replaced by machine processable XML descriptions of the required constraints at
 1228 a later date.

1229
 1230 The buttons at the foot of the Entity Definition form are:

- 1231 - Submit this Entity:
- 1232 - New Entity Definition: Resets the form so that details of another

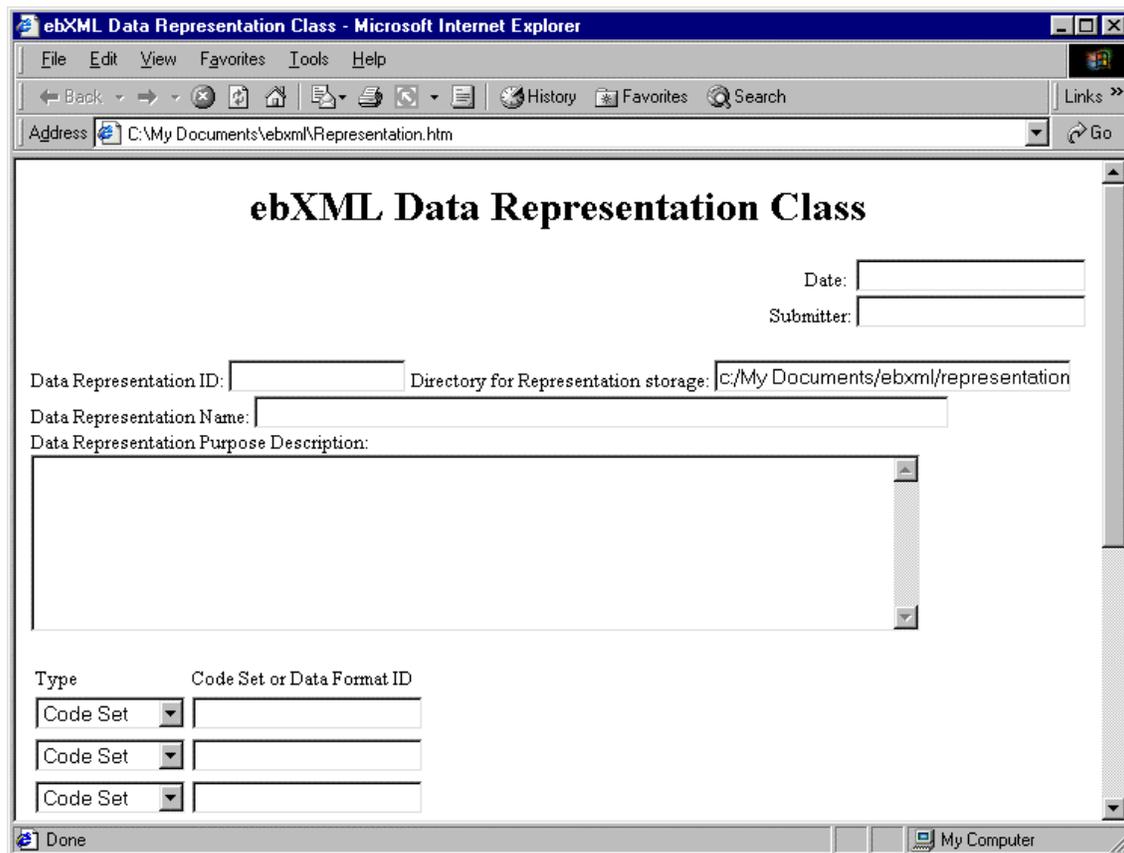
1233 component can be defined. (Each embedded component defined using
 1234 Entity Description in the Type field requires completion of a separate
 1235 Entity Description form.)
 1236 - Create Data Representation Definition: Calls up the form needed to
 1237 record a Data Representation. (Each embedded component defined
 1238 using Data Representation in the Type field requires completion of a
 1239 Data Representation form if the identified format has not previously
 1240 been defined.)
 1241

1242 7.3 Defining a Data Representation

1243
 1244 A data representation identifies one or more data formats and/or code sets that
 1245 can be used to record a particular type of data stored within a core component.
 1246

1247 **Note:** While normally a data representation will only define data of the same type
 1248 (e.g. one or more data formats or one or more code lists) there are cases where
 1249 both a code list and a data format will be required (e.g. to define a list of known
 1250 codes plus a pattern that can be used to extend the list where appropriate.)
 1251

1252 The following figure shows the fields used to define a new data representation for
 1253 use within a core component. The status line at the foot of the form will change as
 1254 you move from field to field to provide you with information about the type of data
 1255 to be entered in each field.



1256
 1257 The fields provided on this form are:
 1258 - Date of Submission: enter an ISO 8601 conformant date using the

- 1259 format CCYY-MM-DD here
- 1260 - Submitting Organization: enter the name of the submitting
- 1261 organization here
- 1262 - Contact E-mail Address: enter e-mail address to which questions can
- 1263 be submitted
- 1264 - Date Representation ID: Enter a unique identifier for the data
- 1265 representation. (This will form the file name of the data representation
- 1266 definition.)
- 1267 - Directory for Representation storage: Should indicate the path
- 1268 required to reach the required directory. (When the default directory
- 1269 set up is being used this entry should not need to be updated from the
- 1270 pre-assigned values.)
- 1271 - Data Representation Name: Name to be used to identify the data
- 1272 representation. (May or may not be the same as the Date
- 1273 Representation ID.)
- 1274 - Data Representation Description: Enter description of role of data
- 1275 representation that allows it to be distinguished from other
- 1276 representations.
- 1277 - Type: Select Code Set if the representation is to reference a code set,
- 1278 or Data Format if it is to reference a data format
- 1279 - Code Set or Data Format ID: Enter the unique identifier to be assigned
- 1280 to the referenced definition.
- 1281

1282 The form ends with the following buttons:

- 1283 - Submit this Representation: Creates XML record of form and displays
- 1284 this in a separate window. (Dismiss window if record is accurate:
- 1285 otherwise return to source and use the Back button to return to your
- 1286 entries so that they can be corrected and resubmitted.)
- 1287 - Define another Representation: Resets the form so that details of
- 1288 another representation can be defined.
- 1289 - Define Code Set: Calls up the form needed to record a Code Set. (Each
- 1290 format defined using Code Set in the Type field requires completion of
- 1291 a Domain Value Code Set form if the identified format has not
- 1292 previously been defined.)
- 1293 - Define Data Format: Calls up the form needed to record a Data
- 1294 Format. (Each format defined using Data Format in the Type field
- 1295 requires completion of a Domain Value Data Foramt form if the
- 1296 identified format has not previously been defined.)
- 1297

1298 **7.4 Defining a Data Format**

1299

1300 The following figure shows the fields used to define a code set for use within a core
 1301 component representation. The status line at the foot of the form will change as
 1302 you move from field to field to provide you with information about the type of data
 1303 to be entered in each field.
 1304

1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329

The fields provided on this form are:

- Date of Submission: enter an ISO 8601 conformant date using the format CCYY-MM-DD here
- Submitting Organization: enter the name of the submitting organization here
- Contact E-mail Address: enter e-mail address to which questions can be submitted
- Data Format ID: Enter a unique identifier for the data format. (This will form the file name of the data format definition.)
- Directory for Data Format Set storage: Should indicate the path required to reach the required directory. (When the default directory set up is being used this entry should not need to be updated from the pre-assigned values.)
- Data Format Concept Name: Enter name to be used to identify the concept behind code list. (May or may not be the same as the Data Format ID.)
- Data Format Concept Description: Enter description of role of data format that allows it to be distinguished from other data formats.
- Data Format Type: If code set has pattern based on one of the known data format languages select relevant entry from list. Otherwise select None
- Character Set: If relevant, select controlling character set from list supplied.
- Data Format Definition: If relevant, enter pattern that defines format

1330 of codes set values using language identified in Code Set Data Format
1331 field

1332

1333 The form ends with the following buttons:

1334 - Submit this Data Format: Creates XML record of form and displays
1335 this in a separate window. (Dismiss window if record is accurate:
1336 otherwise return to source and use the Back button to return to your
1337 entries so that they can be corrected and resubmitted.)

1338 - Define another Data Format: Resets the form so that details of
1339 another code set can be defined.

1340 - Define Code Set: Calls up the form needed to define a Code Set.

1341 - New Entity Definition: Calls up the form needed to define an Entity.

1342 - Create Data Representation Definition Calls up the form needed to
1343 define a Data Representation.

1344

1345 **7.5 Defining a Code Set**

1346

1347 The following figures show the fields used to define a code set for use within a core
1348 component representation. The status line at the foot of the form will change as
1349 you move from field to field to provide you with information about the type of data
1350 to be entered in each field.

1351

1352

ebXML Value Domain Code Set Definition - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Refresh Print Stop History Favorites Search Links

Address C:\My Documents\ebxml\CodeSet.htm Go

ebXML Value Domain Code Set Definition

Date of Submission:

Submitting Organization:

Contact E-mail Address:

Code Set ID: Directory for Code Set storage:

Control Agency: Agency Assigned Name:

Code Set Concept Name:

Code Set Concept Description:

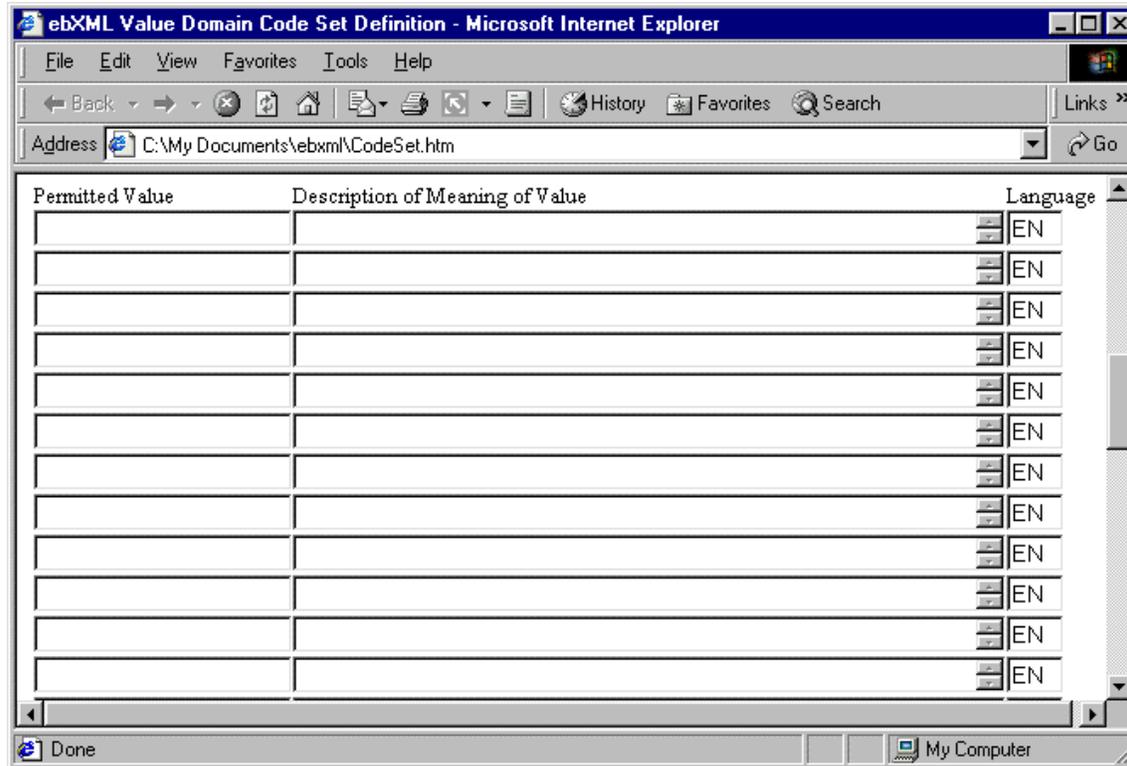
Code Set Data Format Type:

Code Set Character Set:

Code Set Data Format Definition:

Done My Computer

1353



1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382

The fields provided on this form are:

- Date of Submission: enter an ISO 8601 conformant date using the format CCYY-MM-DD here
- Submitting Organization: enter the name of the submitting organization here
- Contact E-mail Address: enter e-mail address to which questions can be submitted
- Code Set ID: Enter a unique identifier for the code set. (This will form the file name of the code set definition.)
- Directory for Code Set storage: Should indicate the path required to reach the required directory. (When the default directory set up is being used this entry should not need to be updated from the pre-assigned values.)
- Code Set Control Agency: Enter name of agency responsible for adding new entries to code set.
- Code Set Concept Name: Enter name to be used to identify the concept behind code list. (May or may not be the same as the Code Set ID.)
- Code Set Concept Description: Enter description of role of code set that allows it to be distinguished from other code sets.
- Code Set Data Format Type: If code set has pattern based on one of the known data format languages select relevant entry from list. Otherwise select None
- Code Set Character Set: If relevant, select controlling character set from list supplied.
- Code Set Data Format Definition: If relevant, enter pattern that defines format of codes set values using language identified in Code Set Data Format field
- Permitted Value: Enter permitted code set values in the fields in this

- 1383 column.
1384 - Description of Meaning of Value: Enter description of how value is to
1385 be interpreted.
1386 - Language: If description is not in English, enter ISO 639 code for
1387 language used for description.
1388

1389 The form ends with the following buttons:

- 1390 - Submit this Code Set: Creates XML record of form and displays this in
1391 a separate window. (Dismiss window if record is accurate: otherwise
1392 return to source and use the Back button to return to your entries so
1393 that they can be corrected and resubmitted.)
1394 - Define another Code Set: Resets the form so that details of another
1395 code set can be defined.
1396 - Define Data Format: Calls up the form needed to record a Data
1397 Format.
1398 - New Entity Definition: Calls up the form needed to define an Entity.
1399 - Create Data Representation Definition Calls up the form needed to
1400 define a Data Representation.
1401

1401

1402 **Troubleshooting**

1403 The forms described in this document have not been fully tested to date. If you
1404 encounter any problems using them please contact their author, Martin Bryan,
1405 via email at mtbryan@sgml.u-net.com (please be patient as I am traveling a lot
1406 during May/June so may not be able to respond as fast as you would like.)
1407

1407

1408