

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 2<sup>nd</sup> project team  
92 meeting of ebXML in Orlando, will be proposed in the paper. This paper will also offer the templates  
93 for describing the results of analysis on the business entities.

94

95 The purposes of the paper includes,

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

99 2. To show how to analyze business entities currently being used in EDI in order to design  
100 the object classes for object modeling experts, who have been engaged in designing the  
101 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 1. Syntactically neutral,

110 2. Conforming to ISO11179 and

111 3. Aligning with the Unified Modeling Methodology.

112 4. A mechanization for developing Core Components

113

114 **Revision note :**

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

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

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

118 In the 3<sup>rd</sup> revision of the draft paper, Representation Class candidates are introduced in the section  
119 3.1.

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

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

123 In the 4<sup>th</sup> revision of the draft paper, the XML based templates are amended through some testing.

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

125 **2. Business entity analysis**

126

127 At the 2<sup>nd</sup> 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 **Party**  
NAD : Names and addresses of the parties relevant to the order/invoice.

139 The qualifier of NAD is specifying the function of the party (seller or buyer)

140 LOC : Specific location information of the party

141

142 FII : Financial institution and relevant account number for the party

143

144 RFF : Reference for the party

145

146 [ DTM : Date and/or time related to the reference

147

148 DOC : Information relating to the documents required by the party specification

149

150 [ DTM : Date and/or time related to the document

151

152 CTA : Person or department whom communications should be directed

153

154 [ COM : Communication type and number for the contact

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 IPOAD (Insurance policy administration message). The typical pattern of the simple  
163 form of 'Party' is as follows.

164

165

166 **Party**  
PNA : Name of the party and their function relevant to the message.

167

168 ADR : Addresses of the party.

169

170 CTA : Person or department whom communications should be directed

171

172 [ COM : Communication type and number for the contact

173

174 \*Note: RFF segment is also used in several messages in addition to the above pattern.

175

176

177 Fig.2 Party example – 2

178

179 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)

183 .

184

185

186

187

188

189

190

191

192

193

194

195

196

197

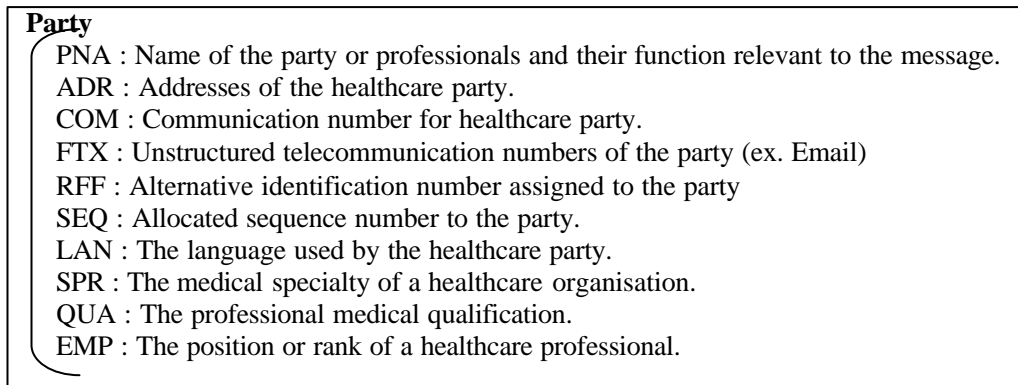


Fig.3 Party example – 3

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

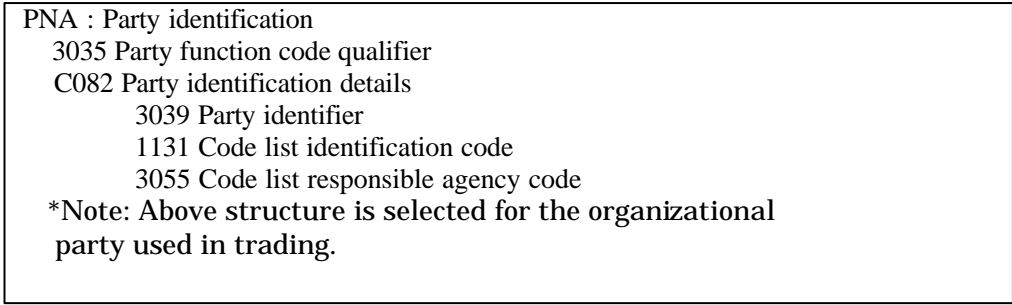


Fig.4 PNA segment structure

(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'.

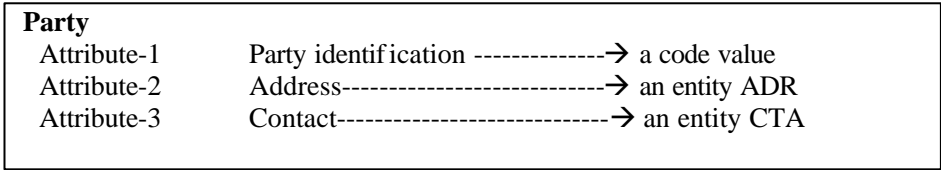


Fig.5 Attributes of Party

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.

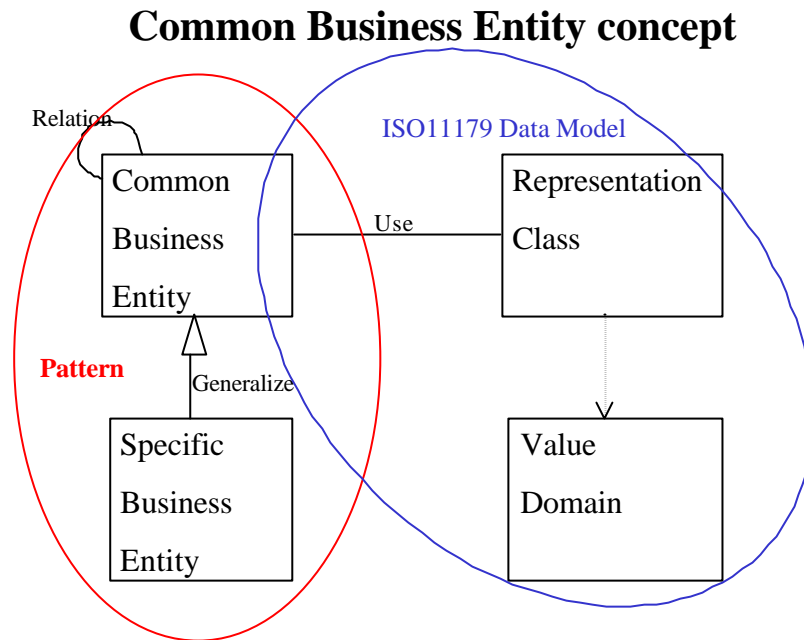
When the data element has the number representation, the value can be any arithmetical value. All the arithmetical values are one of the value domains.

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 follows.



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

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).

302 The following items are kinds of Core Components.

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

308

309 Analysis patterns may be used in the business process.

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

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

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

318

### 319 **3.1 Representation Class candidates**

320

321 The following lists are the candidates for Representation Class.

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

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

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

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

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

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

335 **percent** A rate expressed in hundredths between values that have the  
336 same unit of measure.

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

339 **rate** A quantity or amount measured with respect to another  
340 measured quantity or amount.

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

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

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

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

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

346

### 347 **3.2 Core Component Meta model**

348

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

350

351

352

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

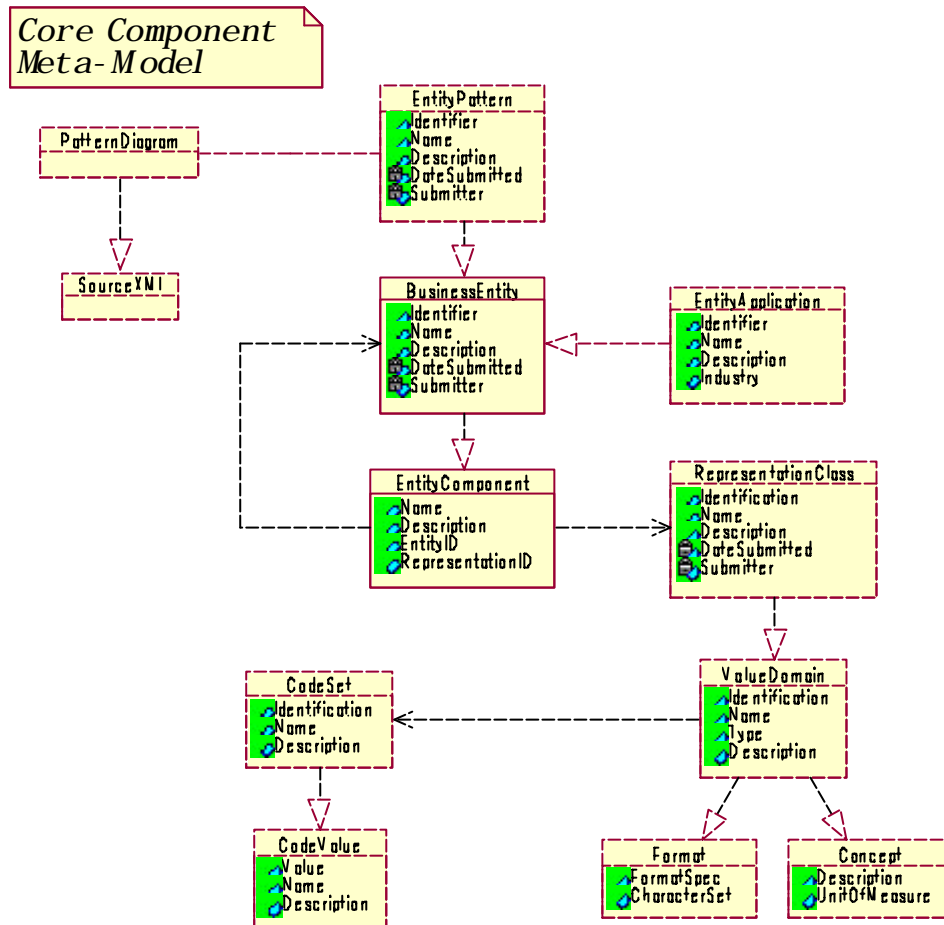


Fig 7 Core Component Meta model

#### 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 certain applications in certain industries, or can be used only between certain trading partners.

Because of the above considerations, we shall select and standardize the proper level of

405 components.

406

## 407 **4.2 Syntactically neutral**

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

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

416

## 417 **4.3 Basic Semantic Register (BSR)**

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

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

431

## 432 **4.4 Naming the component**

433 There are three purposes of naming the component.

434 1. To identify the component uniquely in the certain domain if there are no identifiers other than  
435 the name.

436 2. To be recognized easily by human.

437 3. To specify the domain structure.

438 The data element conforming ISO11179 shall have the unique identifier other than the name,  
439 therefore, the name of the data element is the primary means of identification of objects and concepts  
440 for humans. Otherwise, the name is the only identification for the component in UML. Also the  
441 element name is the only identification for the component in XML. However, an element name in  
442 XML may have different definitions depending upon the context of the element within the XML  
443 structure.

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

447

448

## 449 **4.5 Segmentation of Core Components**

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

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

455 “A role name defines a task or duty of a class in an association with another class.” The example  
456 they use is the qualifier of the Party segment that indicates whether the party concerned is the Buyer,  
457 Seller, Consignor, Dispatch Party, etc.

458 The Pharos document also contains a number of “rules” for the creation of UML models to represent

459 business messages. Rule 5 states:

460

461

“Pharos Rule 5: Create multiple associations

462

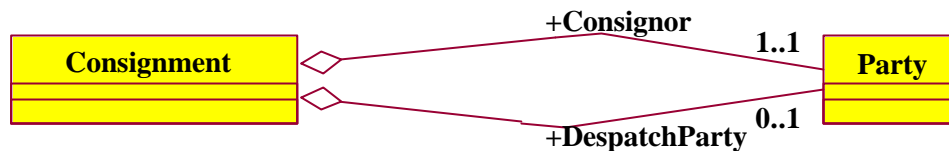
463

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

464

465

466



467

468

469

Fig. 8 Example of multiple associations.

470

471

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

472

473

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

474

475

476

477

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

478

A Consignment has minimum 1 and maximum 1 Consignor

479

A Consignment has minimum 0 and maximum 1 DispatchParty”

480

481

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

482

483

484

485

486

487

488

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

489

490

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

491

492

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

493

494

495

496

497

498

499

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

500

501

502

503

Order

504

Parties

505

Seller

506

Party Information

507

Party Identification

508

....

509

510 In discussions of the UK Data Harmonization Group at e-centre<sup>uk</sup> it was pointed out that the Parties  
511 and Party Information component of this sequence were really just containers that allowed the  
512 correct management of data that forms part of the same abstract class. The other three components of  
513 the “tree” represent the Business Process being undertaken (Order), the association of the abstract  
514 class containing party information with the business process (Seller), and the core components that  
515 identify the information to be interchanged for the completion of that part of the business process  
516 (Party Identification ...).

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

523

## 524 **5. Describe the Core Component**

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

528

529 (1) Describing the business entity pattern

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

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

551

552 (2) Describing representation class

553 Each representation class can be described in one sheet.

- 554 - Representation class definition
  - 555 Representation class ID
  - 556 Representation class name
  - 557 Representation class description
  - 558 Data type definitions
    - 559 Concept (lexical meanings)
      - 560 Concept ID
      - 561 Concept specification
      - 562 Unit of Measure if needed

563                                   Format (syntactical expressions)  
564                                    Format ID  
565                                    Format specification  
566                                    Character set if needed  
567                   Value domain list  
568                                    Value domain ID  
569                                    Value domain name  
570                                    Value specification in the domain  
571  
572

573 6. Sample description

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

<b>Pattern Definition</b>	
	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.	

589 **Class diagram**

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

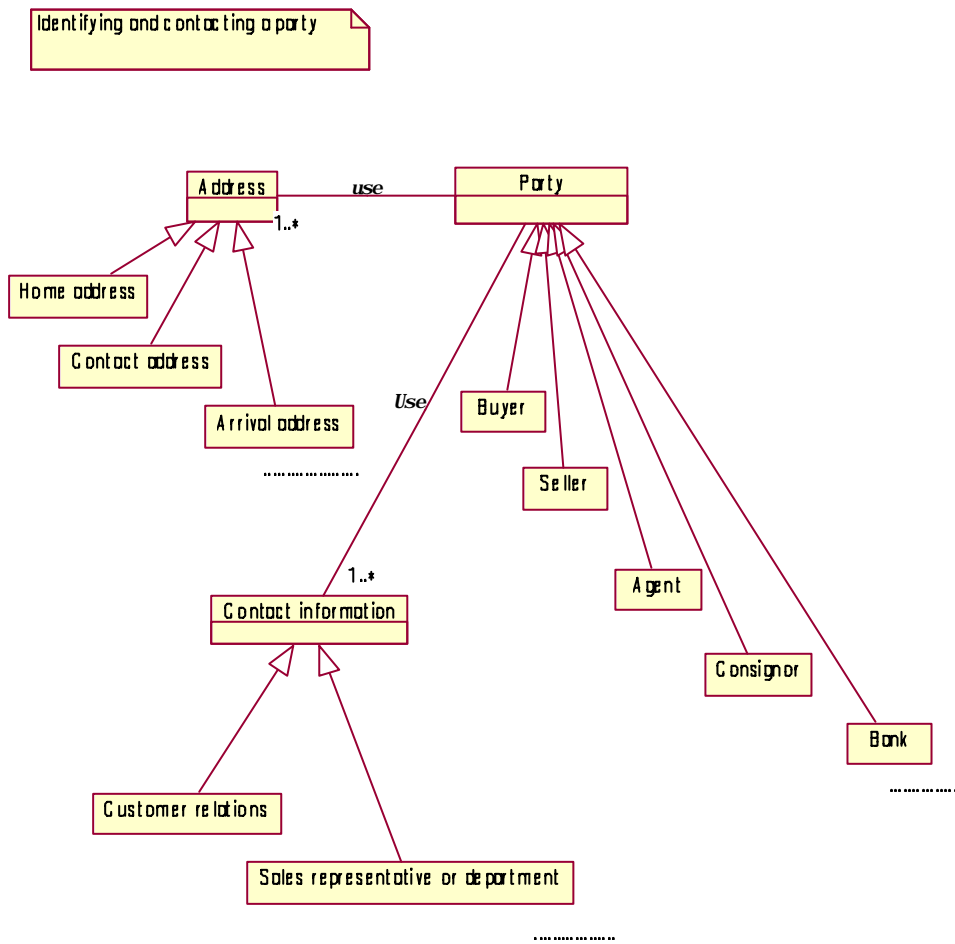
619

620

621

622

623



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

624  
 625  
 626  
 627  
 628  
 629  
 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

## 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.

676  
677  
678  
679  
680  
681  
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

## 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.

729  
730  
731  
732  
733  
734  
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

## 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.

782  
 783  
 784  
 785  
 786  
 787  
 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

**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
3 .....etc	
*The identifier schemas are defined in the code list of data element 1131/3055 in UN/EDIFACT directory.	

835  
836  
837  
838  
839  
840  
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

**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

887  
888  
889  
890  
891  
892  
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

## 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

941  
 942  
 943  
 944  
 945  
 946  
 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

## 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.	

994  
 995  
 996  
 997  
 998  
 999  
 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

## 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.	

## 1046 **7. Instructions for capturing ebXML Core Component definitions**

1047

1048 The following instructions describe how Microsoft's Internet Explorer 5.0 (IE5) can be  
1049 used to capture core component definitions as XML files.

1050

1051 **WARNING:** This process will not work on Netscape Explorer or on older versions of  
1052 Internet Explorer as it relies on Microsoft specific extensions to HTML.

1053

1054 The UML metamodel for ebXML core components recognizes the following types of  
1055 record:

1056

1057 - **Patterns:** UML models that define a set of related data entities, and the  
1058 associations between them

1059 - **Entities:** UML model components that identify sets of related components,  
1060 including data elements with a specific data representation

1061 - **Date Representations:** the set of data patterns or code lists that can be  
1062 used to capture a particular component of an entity

1063 - **Code Sets:** A list of permitted values that can be used to complete a  
1064 component, together with descriptions of their meaning

1065 - **Data Formats:** Details of constraints to be placed on the contents  
1066 particular components of a pattern.

1067

1068 The HTML forms provided in this suite allow each of these data types to be recorded.  
1069 To allow the relationships between forms to be clearly identified the basic UML model  
1070 has been extended by adding elements that reference the ID of the next lowest level of  
1071 component in the model. (This means that once a component, data representation, code  
1072 list or data format has been defined once it need only be referenced in subsequent  
1073 models.)

1074

1075 For each of the above record types there is a form with the appropriate name  
1076 (pattern.htm, entity.htm, representation.htm, CodeSet.htm and DataFormat.htm).  
1077 Each form contains buttons that allow the current contents to be submitted for storage,  
1078 reset so that a new entry of the same type can be made or request a form for a record at  
1079 a lower level in the metamodel.

1080

1081 The package has been designed to be run within a directory called ebxml within the My  
1082 Documents section of your C drive. If wish to use another drive or root directory you  
1083 should search each of the above files for any occurrences of the string "C:/My  
1084 Documents/ebxml/" and replace this with the appropriate identifier for the directory  
1085 you wish to use.

1086

1087 Within the directory that is used to contain the forms you will need to create five  
1088 subdirectories prior to using the forms. These subdirectories should be labelled  
1089 patterns, entities, representations, codesets and dataformats respectively. A further  
1090 directory, called class, can also be defined to store class diagrams, etc. (Directories with  
1091 these names may or may not have been created for you when you unzipped the source  
1092 files, depending on your settings of your file unzipper.)

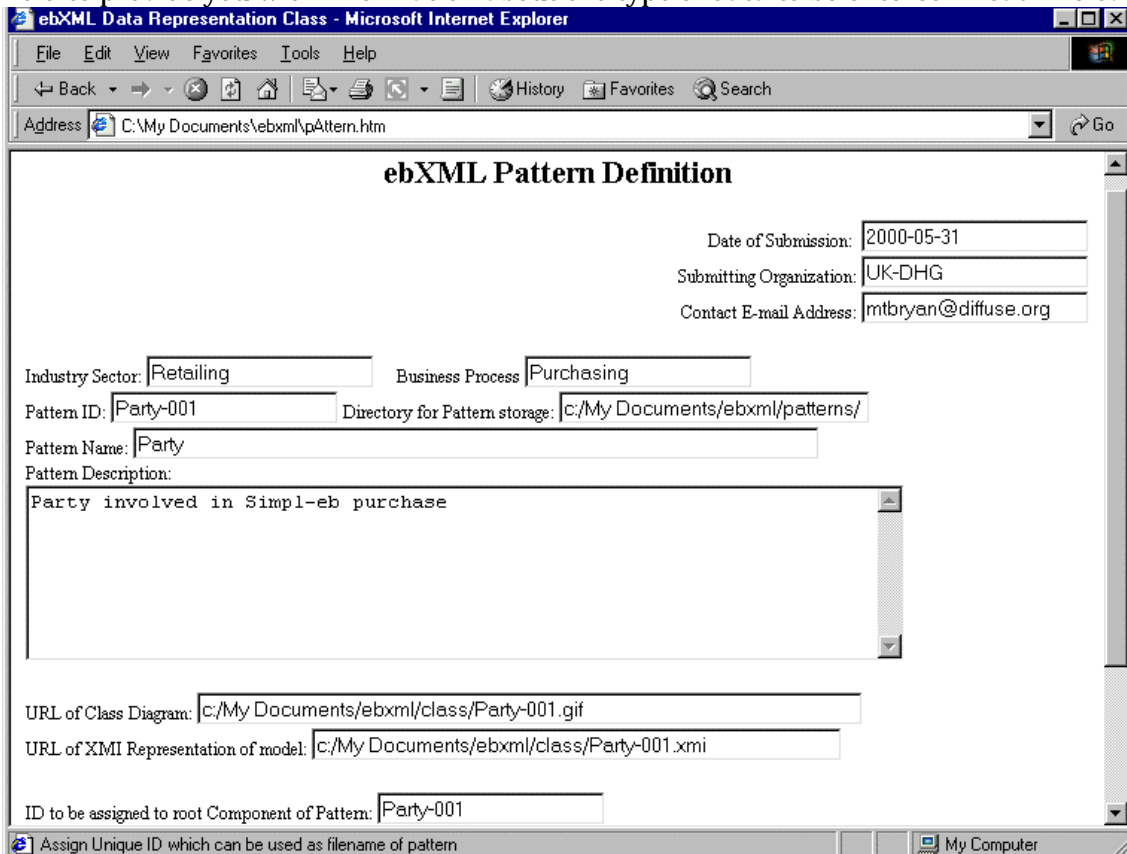
1093

1094 Because the forms submit their contents to the local file store rather than to a remote  
1095 directory you will need to ensure that IE5 has been set up to permit this. In the Tools  
1096 Menu select the Internet Options entry and then the Security tab. If your security level  
1097 is High you will not be able to use these forms. If it is set to Medium you will be asked

1098 to confirm that each record may be written to disc. If it is Low you will be able to write  
 1099 the files without having to confirm each one, but must take care if using the Internet. (I  
 1100 find that the Medium level, which requires me to confirm before writing but still  
 1101 provides for a safe level of Internet access is an adequate compromise.)

1102  
 1103 **7.1 Defining a Pattern**

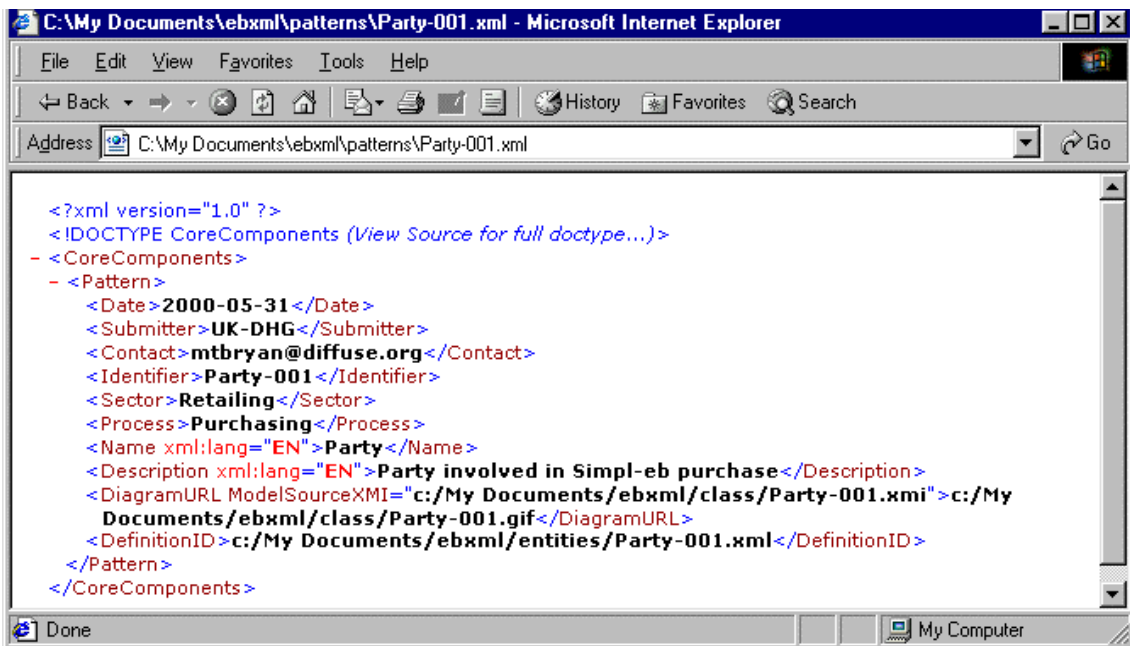
1104  
 1105 The following figure shows the fields used to define a new pattern for use as a core  
 1106 component. The status line at the foot of the form will change as you move from field to  
 1107 field to provide you with information about the type of data to be entered in each field.



- 1108  
 1109 The fields provided are:
- 1110 - Date of Submission: enter an ISO 8601 conformant date using the format
  - 1111 CCYY-MM-DD here
  - 1112 - Submitting Organization: enter the name of the submitting organization
  - 1113 here Contact E-mail Address: enter e-mail address to which questions can
  - 1114 be submitted
  - 1115 - Industry Sector: identify which communities the pattern is expected to be
  - 1116 used by (using SIC codes where available)
  - 1117 - Business Process: record business process(es) to which pattern applies
  - 1118 (using SIC codes where available)
  - 1119 - Pattern ID: Enter a unique identifier for the pattern. (This will form the
  - 1120 file name of the pattern definition.)
  - 1121 - Directory for pattern storage: Should indicate the path required to reach
  - 1122 the required directory. (When the default directory set up is being used
  - 1123 this entry should not need to be updated from the pre-assigned values.)

- 1124 - Pattern Name: Name to be used to identify the pattern. (May or may not
- 1125 be the same as the Pattern ID.)
- 1126 - Pattern Description: Enter description of role of pattern that allows it to be
- 1127 distinguished from other patterns.
- 1128 - URL of Class Diagram: Enter URL of file containing printable version of
- 1129 class diagram. (For ease of use on the web we recommend this be a GIF
- 1130 file. If this is stored in the class subdirectory within you ebxml directory
- 1131 then all you need to do is to replace the ??? in the default name displayed
- 1132 with the filename.)
- 1133 - URL of XMI Representation of model: If an XMI representation of the
- 1134 model is available for interchange enter the appropriate file reference
- 1135 here: otherwise delete the default value.
- 1136 - ID to be assigned to root Component of Pattern: Indicate which ID you
- 1137 expect to assign to the Entity which will form the root of the pattern
- 1138 - Directory that will be used for Entity storage: Should indicate the path
- 1139 required to reach the required directory. (When the default directory set
- 1140 up is being used this entry should not need to be updated from the pre-
- 1141 assigned values.)
- 1142

1143 When all relevant fields have been completed click on the Submit this Pattern button.  
 1144 This will cause the inbuilt program to store an XML record pattern and then display  
 1145 the contents of the file as shown below.



1146  
 1147 Once you have ascertained that this file correctly records the details of your pattern the  
 1148 window can be dismissed from the screen.

1149  
 1150 The Pattern form ends with a button that allows you to "Create Entity Definition for  
 1151 Root Component". Clicking on this button will call up the form needed to record each of  
 1152 the entities defined within the pattern.

1153  
 1154 **7.2 Defining an Entity**

1155

1156 The following figures show the fields used to define a new entity for use within a core  
1157 component. The status line at the foot of the form will change as you move from field to  
1158 field to provide you with information about the type of data to be entered in each field.

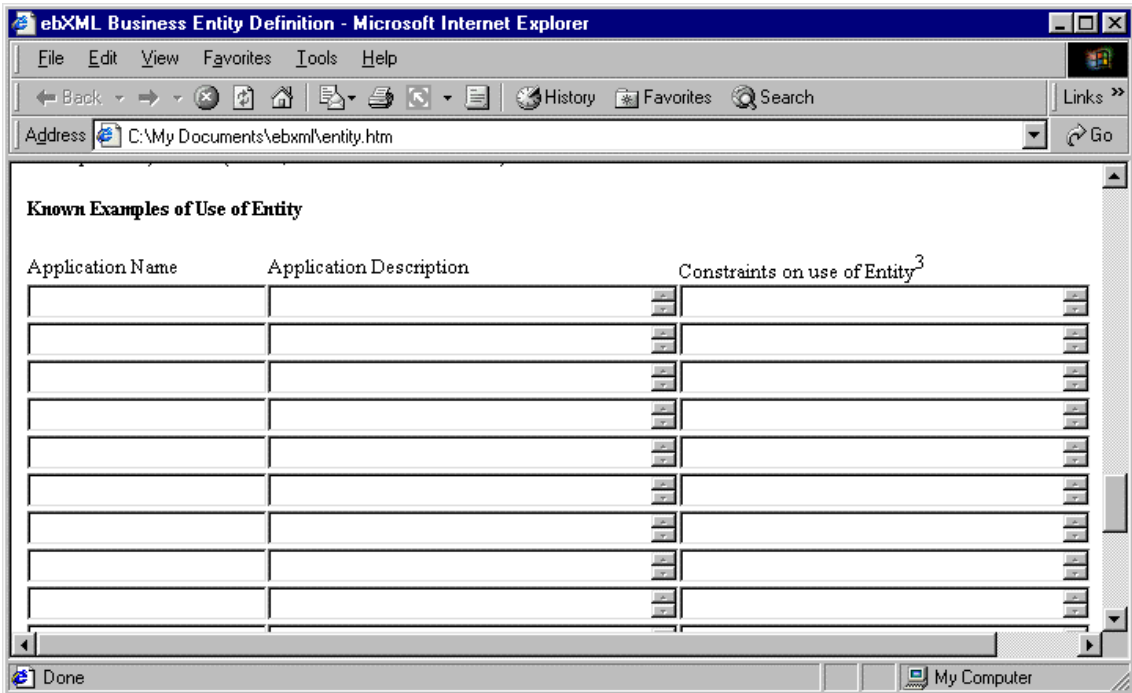
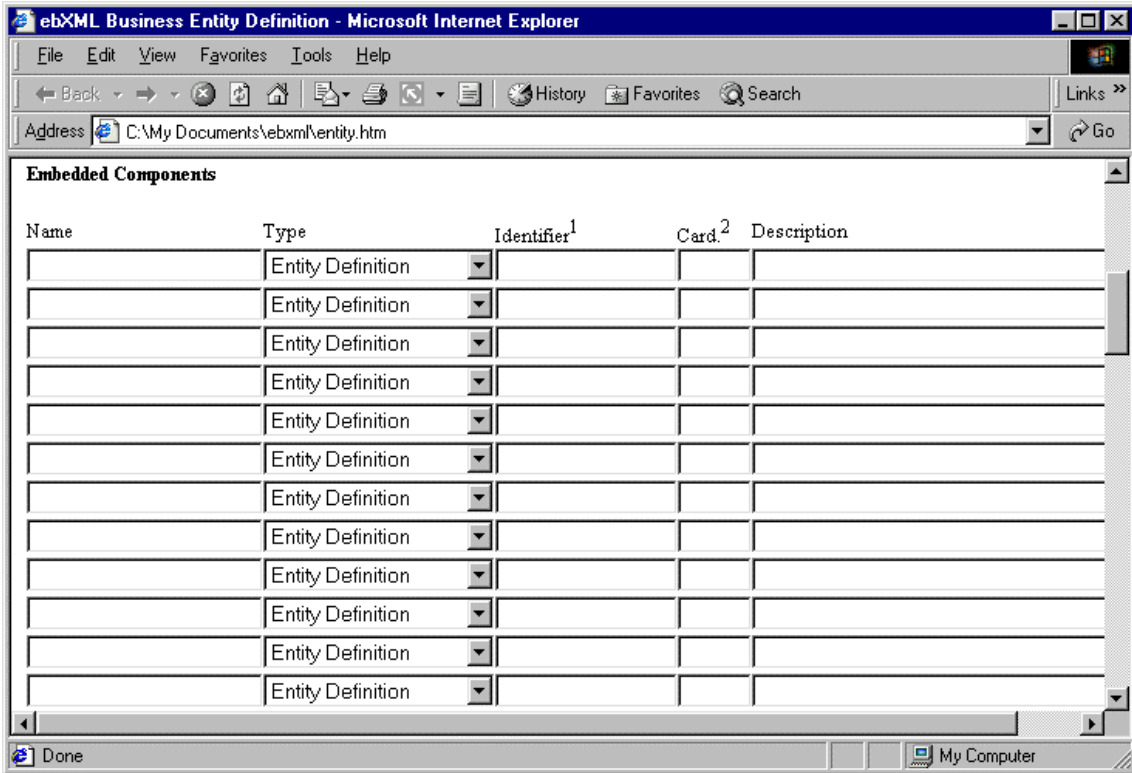
The screenshot shows a Microsoft Internet Explorer window titled "ebXML Business Entity Definition - Microsoft Internet Explorer". The address bar displays "C:\My Documents\ebxml\entity.htm". The main content area features the title "ebXML Business Entity Definition" in a large, bold font. Below the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

1159

1160



1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167

- The fields provided in the first part of the 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

- 1168 - Contact E-mail Address: enter e-mail address to which questions can be
- 1169 submitted
- 1170 - Industry Sector: identify which communities the pattern is expected to be
- 1171 used by (using SIC codes where available)
- 1172 - Business Process: record business process(es) to which pattern applies
- 1173 (using SIC codes where available)
- 1174 - Entity ID: Enter a unique identifier for the entity. (This will form the file
- 1175 name of the entity definition.)
- 1176 - Directory for entity storage: Should indicate the path required to reach the
- 1177 required directory. (When the default directory set up is being used this
- 1178 entry should not need to be updated from the pre-assigned values.)
- 1179 - Entity Name: Name to be used to identify the entity. (May or may not be
- 1180 the same as the Entity ID.)
- 1181 - Entity Description: Enter description of role of entity that allows it to be
- 1182 distinguished from other core components.
- 1183

1184 Each line in the section headed Embedded Components has the following fields:

- 1185 - Name: Name of component in form suitable for use in XML DTD (i.e.
- 1186 starting with a letter and containing no spaces)
- 1187 - Type: Type of component (either Entity Definition if there are embedded
- 1188 components or Data Representation if this component is a root one
- 1189 designed to transfer data between systems)
- 1190 - Identifier: Indicate which ID you expect to assign to the
- 1191 Entity/Representation when you define it (this will create a cross reference
- 1192 to the definition you will create at a subsequent stage using either another
- 1193 copy of this form or the form for recording data representations.)
- 1194 - Card. If the embedded component is optional and/or repeatable the
- 1195 cardinality of this component with respect to its parent (i.e. how many
- 1196 times it can occur within the parent) as defined in the UML model should
- 1197 be recorded here, expressed as 0..1 (optional), 0..\* (optional and
- 1198 repeatable), 1..\* (required and repeatable) or m..n (m=min, n=max no of
- 1199 occurrences).
- 1200 - Description: Brief description of role of component within Entity.
- 1201

1202 If the entity is a pattern that is intended to be sub-classed or used as an abstract class  
 1203 that is associated with concrete classes, examples of the intended use of the entity can  
 1204 be recorded in the section headed Known Examples of Use of Entity. The following  
 1205 fields can be used for each example:

- 1206 - Application Name: Name by which sub-class or association is known
- 1207 - Application Description: Description of purpose of application
- 1208 - Constraints on use of Entity: Details of any constraints that apply to the
- 1209 use of the entity within this application (e.g. components that must or may
- 1210 not be used for this application of the pattern/entity.)
- 1211

1212 **Note:** At present there is no formal language for defining such constraints but in  
 1213 future it is anticipated that text based descriptions entered initially will be replaced by  
 1214 machine processable XML descriptions of the required constraints at a later date.

1215  
 1216 The buttons at the foot of the Entity Definition form are:

- 1217 - Submit this Entity:
- 1218 - New Entity Definition: Resets the form so that details of another
- 1219 component can be defined. (Each embedded component defined using

- 1220 Entity Description in the Type field requires completion of a separate
- 1221 Entity Description form.)
- 1222 - Create Data Representation Definition: Calls up the form needed to record
- 1223 a Data Representation. (Each embedded component defined using Data
- 1224 Representation in the Type field requires completion of a Data
- 1225 Representation form if the identified format has not previously been
- 1226 defined.)

### 7.3 Defining a Data Representation

1229 A data representation identifies one or more data formats and/or code sets that can be  
 1230 used to record a particular type of data stored within a core component.

1231 **Note:** While normally a data representation will only define data of the same type (e.g.  
 1232 one or more data formats or one or more code lists) there are cases where both a code  
 1233 list and a data format will be required (e.g. to define a list of known codes plus a  
 1234 pattern that can be used to extend the list where appropriate.)

1235 The following figure shows the fields used to define a new data representation for use  
 1236 within a core component. The status line at the foot of the form will change as you  
 1237 move from field to field to provide you with information about the type of data to be  
 1238 entered in each field.

The screenshot shows a web browser window titled "ebXML Data Representation Class - Microsoft Internet Explorer". The address bar shows "C:\My Documents\ebxml\Representation.htm". The main content area displays the "ebXML Data Representation Class" form. The form includes the following fields:

- Date:
- Submitter:
- Data Representation ID:  Directory for Representation storage:
- Data Representation Name:
- Data Representation Purpose Description:

At the bottom of the form, there is a table with two columns: "Type" and "Code Set or Data Format ID".

Type	Code Set or Data Format ID
Code Set	<input type="text"/>
Code Set	<input type="text"/>
Code Set	<input type="text"/>

- 1242 The fields provided on this form are:
- 1243 - Date of Submission: enter an ISO 8601 conformant date using the format
  - 1244

- 1245 CCYY-MM-DD here
- 1246 - Submitting Organization: enter the name of the submitting organization
- 1247 here
- 1248 - Contact E-mail Address: enter e-mail address to which questions can be
- 1249 submitted
- 1250 - Date Representation ID: Enter a unique identifier for the data
- 1251 representation. (This will form the file name of the data representation
- 1252 definition.)
- 1253 - Directory for Representation storage: Should indicate the path required to
- 1254 reach the required directory. (When the default directory set up is being
- 1255 used this entry should not need to be updated from the pre-assigned
- 1256 values.)
- 1257 - Data Representation Name: Name to be used to identify the data
- 1258 representation. (May or may not be the same as the Date Representation
- 1259 ID.)
- 1260 - Data Representation Description: Enter description of role of data
- 1261 representation that allows it to be distinguished from other
- 1262 representations.
- 1263 - Type: Select Code Set if the representation is to reference a code set, or
- 1264 Data Format if it is to reference a data format
- 1265 - Code Set or Data Format ID: Enter the unique identifier to be assigned to
- 1266 the referenced definition.

1267  
1268 The form ends with the following buttons:

- 1269 - Submit this Representation: Creates XML record of form and displays this
- 1270 in a separate window. (Dismiss window if record is accurate; otherwise
- 1271 return to source and use the Back button to return to your entries so that
- 1272 they can be corrected and resubmitted.)
- 1273 - Define another Representation: Resets the form so that details of another
- 1274 representation can be defined.
- 1275 - Define Code Set: Calls up the form needed to record a Code Set. (Each
- 1276 format defined using Code Set in the Type field requires completion of a
- 1277 Domain Value Code Set form if the identified format has not previously
- 1278 been defined.)
- 1279 - Define Data Format: Calls up the form needed to record a Data Format.
- 1280 (Each format defined using Data Format in the Type field requires
- 1281 completion of a Domain Value Data Foramt form if the identified format
- 1282 has not previously been defined.)

#### 1283 1284 **7.4 Defining a Data Format**

1285  
1286 The following figure shows the fields used to define a code set for use within a core  
1287 component representation. The status line at the foot of the form will change as you  
1288 move from field to field to provide you with information about the type of data to be  
1289 entered in each field.  
1290

1291  
 1292  
 1293  
 1294  
 1295  
 1296  
 1297  
 1298  
 1299  
 1300  
 1301  
 1302  
 1303  
 1304  
 1305  
 1306  
 1307  
 1308  
 1309  
 1310  
 1311  
 1312  
 1313  
 1314

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

1315

1316 The form ends with the following buttons:

- 1317 - Submit this Data Format: Creates XML record of form and displays this in
- 1318 a separate window. (Dismiss window if record is accurate; otherwise
- 1319 return to source and use the Back button to return to your entries so that
- 1320 they can be corrected and resubmitted.)
- 1321 - Define another Data Format: Resets the form so that details of another
- 1322 code set can be defined.
- 1323 - Define Code Set: Calls up the form needed to define a Code Set.
- 1324 - New Entity Definition: Calls up the form needed to define an Entity.
- 1325 - Create Data Representation Definition Calls up the form needed to define
- 1326 a Data Representation.

1327

### 1328 **7.5 Defining a Code Set**

1329

1330 The following figures show the fields used to define a code set for use within a core  
1331 component representation. The status line at the foot of the form will change as you  
1332 move from field to field to provide you with information about the type of data to be  
1333 entered in each field.

1334

1335

ebXML Value Domain Code Set Definition - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Refresh Print 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:

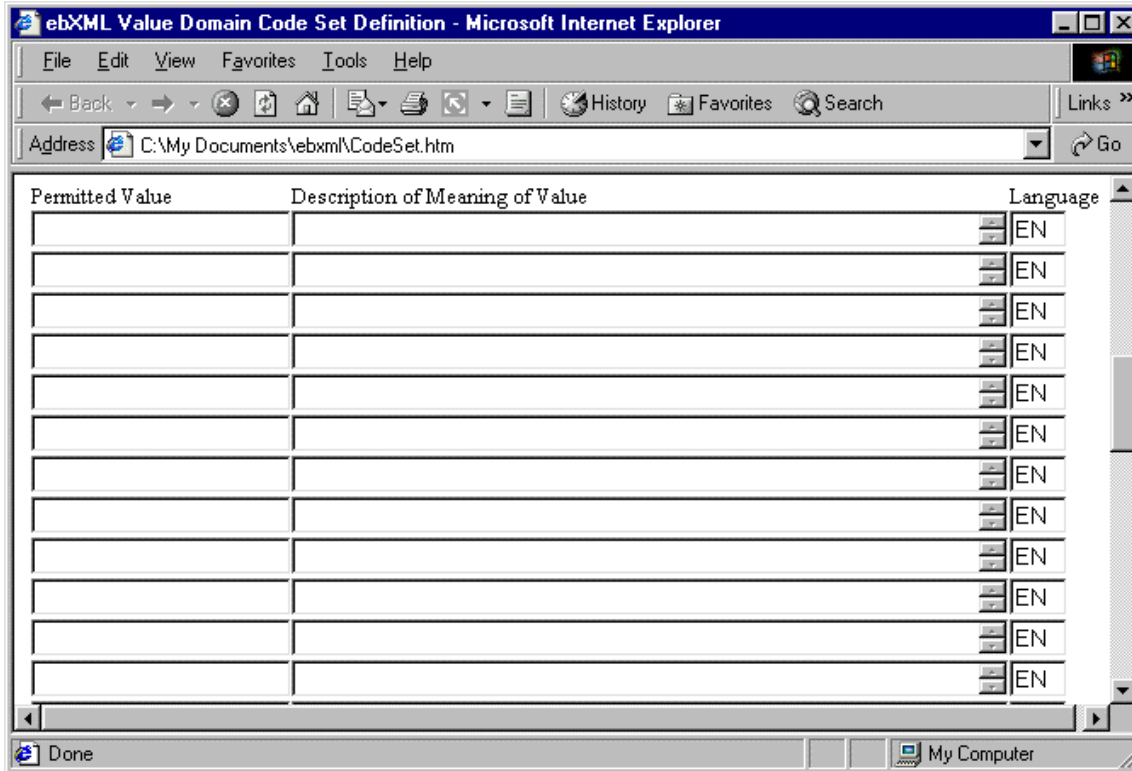
Code Set Data Format Type:

Code Set Character Set:

Code Set Data Format Definition:

Done My Computer

1336



1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362  
 1363  
 1364  
 1365

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 column.

- 1366 - Description of Meaning of Value: Enter description of how value is to be  
1367 interpreted.  
1368 - Language: If description is not in English, enter ISO 639 code for language  
1369 used for description.

1370

1371 The form ends with the following buttons:

- 1372 - Submit this Code Set: Creates XML record of form and displays this in a  
1373 separate window. (Dismiss window if record is accurate: otherwise return  
1374 to source and use the Back button to return to your entries so that they  
1375 can be corrected and resubmitted.)  
1376 - Define another Code Set: Resets the form so that details of another code  
1377 set can be defined.  
1378 - Define Data Format: Calls up the form needed to record a Data Format.  
1379 - New Entity Definition: Calls up the form needed to define an Entity.  
1380 - Create Data Representation Definition Calls up the form needed to define  
1381 a Data Representation.

1382

### 1383 **Troubleshooting**

1384 The forms described in this document have not been fully tested to date. If you  
1385 encounter any problems using them please contact their author, Martin Bryan, via  
1386 email at [mtbryan@sgml.u-net.com](mailto:mtbryan@sgml.u-net.com) (please be patient as I am traveling a lot during  
1387 May/June so may not be able to respond as fast as you would like.)

1388

1389