7 Annex 1 – Additional supporting materials for the Frameworks Integration Guideline

(Non-normative)

(NOTE: the parts in Times New Roman require still significant amount of work – both editing and conceptual. The parts in Arial seem to be mostly OK… The notes in italics mark the areas requiring additions and discussions.)

1. Business Context Matching

Business Context Matching	
Input	Traditional business knowledge, legal agreements between partners, industry specific rules, legal constraints, specific business goals, common business practices and codes of conduct
Output	Two Business Context Models for the integration scenario, defined in a set of UML diagrams (class, collaboration, activity), and an analysis of their matching (and any additional requirements on which the matching depends).
Alternative Procedures	
REA	REA ontology [REA], [REAont]
UMM	Business Requirements View in Chapter 9.2 of [UMM] (can be considered a specialized subset of REA)
EbXML	Business Process Analysis Worksheets and Guidelines [bpWS] (which are also based on REA principles)
SimpleREA	Described below.

1.1. Creating Business Context Models

Simple REA
Here we describe a simplified procedure useful for modeling of simple business cases (based on subset of REA, with relationships to UMM BRV and BTV; it should also be compatible with ebXML). As a result of the pragmatic process described below, you will create an economic exchange diagram, which provides a high-level overview of the parties involved in the business activities; and a value-chain diagram which puts this exchange in a context of the whole enterprise.

1. **Economic Exchange Diagram**

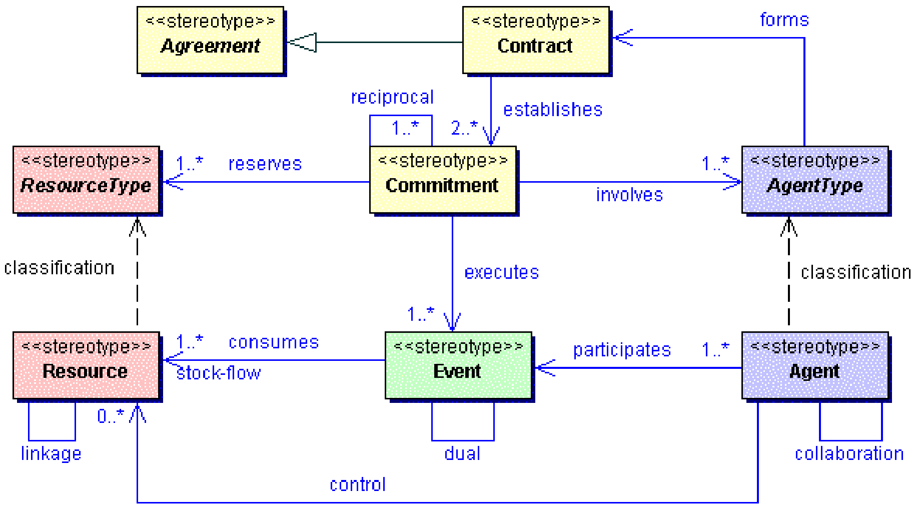
   1.1. Meta-model

   Describe the entities involved in the business case at hand, using the following terms (represented as UML stereotypes):

   - ***AgentType***: the role that a business partner plays in the scenario (e.g. buyer, seller, payer etc…). This is an abstract classification of the concrete Agents involved.
   - ***Agent***: if needed, specifies a concrete representative of a business party, which fulfills a given partner type (e.g. a sales clerk [= seller], a customer [= buyer]).
   - ***Agreement***: an agreement is an arrangement between two partner types that specifies in advance the conditions under which they will trade (terms of shipment, terms of payment, collaboration scenarios, etc.) A special kind of agreement (contract) commits partners to execute specific events, in which economic resources are exchanged.
   - ***Commitment***: an obligation to perform an economic event (i.e. transfer ownership of a specified quantity of a specified economic resource type) at some future point in time.
   - ***EventType***: an abstract classification or definition of an economic event. E.g. rental, service order, direct sales, production (of goods from raw materials), etc …
   - ***Event***: an economic event is the transfer of control of an economic resource from one partner type to another partner type. Examples would include the concrete sales, cash-payments, shipments, leases, deliveries etc. Economic Events usually cause changes in the state of each partner type (so called business events). Therefore they are directly related to (and determine) the transaction boundaries.
   - ***ResourceType***: an economic resource type is the abstract classification or definition of an economic resource. For example, in an ERP system, ItemMaster or ProductMaster would represent the Economic Resource Type that abstractly defines an Inventory item or product. Forms of payment are also defined by economic resource types, e.g. currency.
   - ***Resource***: if needed, specifies a quantity of something of value that is under the control of an enterprise, which is transferred from one partner type to another in economic events. Examples are cash, inventory, labor service and machine service. Contracts deal with resource types (abstract definitions), whereas events deal with resources (real entities). You may use this distinction if needed.

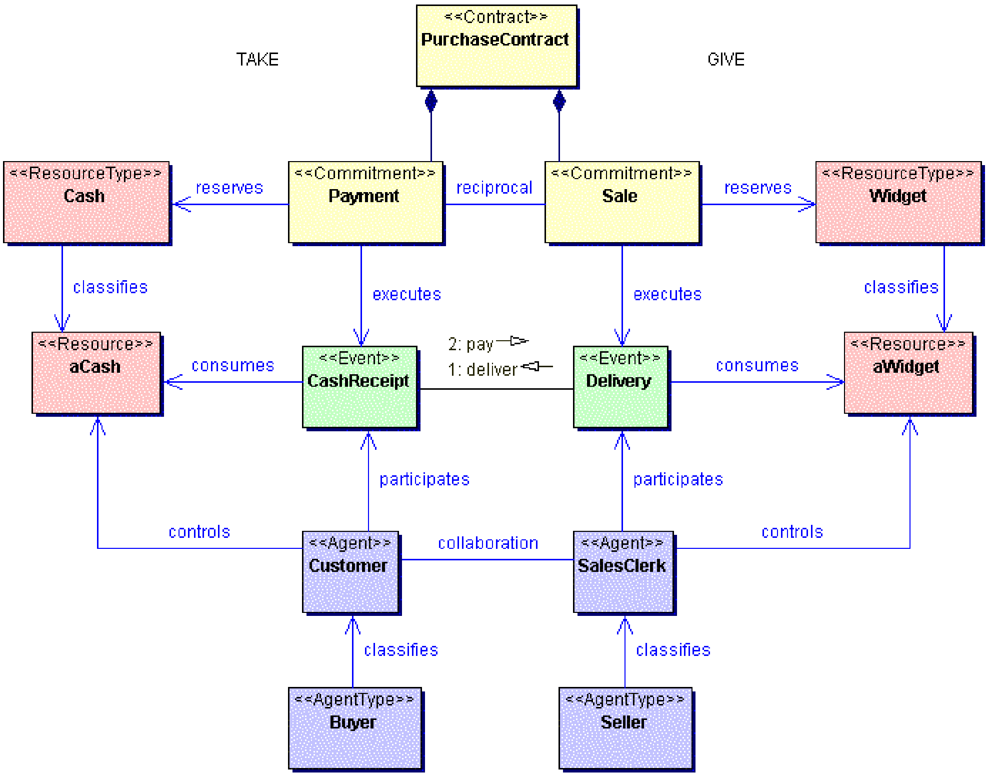
1.2. Meta-model and constraints

The meta-model for building the economic exchange diagrams is presented on the figure below:



The entities have been color-coded. The collaboration between Agents is realized with the BusinessTasks (collaboration protocol), which may be represented as UML activity diagrams.

1.3. Model example



The coloring schema on this diagram corresponds to that on the meta-model diagram.

Note: this diagram shows instances (concrete entities) of types specified above in the meta-model diagram. This is indicated by the UML stereotypes (labels in guillemots). Notice the two messages exchanged in this model – the first is to deliver, the second to pay (but it may be the other way around – an advance payment). This diagram helps us to identify the business transactions (in this case: {deliver, pay}), and also shows us the timing constraints (in this case: first deliver, then pay).

(NOTE: any useful real-life scenario would be more complicated. It could e.g. contain a catalog lookup, negotiation, shipment, blanket agreement, etc… This diagram serves therefore only as an illustration of the approach).

1.2. Checking the Business Context Rules

7.1.1.1.1 #1 Complementary roles

Parties need to play complementary roles (e.g. buyer/seller)

7.1.1.1.2 #2 Expected resources

The resources expected in the exchanges need to be equivalent to the ones expected by the other partner (e.g. cash for goods)

7.1.1.1.3 #3 Timing constraints

The timing constraints on events (commitment specification) need to be mutually satisfiable (e.g. down payment vs. final payment)

7.1.1.1.4 #4 Transaction boundaries

The sequence of expected business transactions needs to be the same (even though the individual business actions may differ)

2. Business Process Mediation

Business Process Mediation	
Input	Business Context models, other information on business processes supporting the business context.
Output	Business Process Models, Business Process Mediator Model for the integration scenario, defined in a set of diagrams (activity/business process, ECIMF process mediation diagram)
Alternative Procedures	
UMM + ECIMF-PM	UMM-BOV, and the ECIMF Process Mediation Model
UML-EDOC + ECIMF-PM	UML-EDOC, and the ECIMF Process Mediation Model
EbXML + ECIMF-PM	Business Process Specification Schema, and the ECIMF Process Mediation Model

2.1. Creating the Business Process Models

(to be completed...)

2.2. Creating the Business Process Mediation model

2.2.1. Check the Business Tasks alignment

- Identify **request and response messages**. *(NOTE: this step will benefit from information collected in BOV and FSV models, if available (cf. [UMM]))*
- Determine **legal obligations** boundaries: which interactions and messages bring what legal and economical consequences. This can be established based on the relationship to the business context diagram. *(NOTE: needs more substance…)*
- Determine the **business transaction boundaries**, rollback (compensation) activities and messages for failed transactions. The transaction boundaries can be better identified with the help of the business context diagram. *(NOTE: needs more substance…)*
- **Identify the differences** in message flow, by comparing message flows between requesting/responding parties for each business task.

2.2.2. Create the Mediation Elements between Business Tasks

- **Missing messages/elements**: are those that are present in e.g. Framework 1 business task B_x (we use the notation $F_1(B_x)$ for that), but don't occur in the corresponding $F_2(B_y, B_z, \ldots)$. This is also true about the individual data elements, which may become available only after certain steps in the conversations, different for each framework. These messages and data elements will have to be created by the mediator, based on already available data from various sources, such as:
  - previous messages
  - configuration parameters
  - external resources

  and sent according to the expected conversation pattern.
- **Superfluous or misplaced messages/elements**: are those that don't correspond directly to any of the required/expected messages as specified in the other framework. Also, they may be required to arrive in different order. The mediator should collect them (for possible use of information elements they contain at some later stage) without sending them to the other party, or change the order in which they are sent. The business context diagram will help determine what kind of re-ordering is possible without breaking the transaction boundaries (it should be possible to change the order within the transaction boundaries without breaking their semantics, but not across them).
- **Different constraints** (time, transactional, legal…): this issue is similar in complexity to resolving the semantic conflicts (see below), and a similar approach could be taken.
  (NOTE: namely???)

3. Semantic translation *(to be completed)*

(NOTE: needs to be harmonized with the methodology section!!!)

- **Identify the key concepts** in use for message exchanges conducted according to each framework, within the context of the selected corresponding business tasks:
  - **For each message** in B_i identify the key indispensable information elements that decide about the success of the information exchange from the business point of view in each of the frameworks:

    $M_i(E_1, E_2, \ldots, E_n)$
  - **For each message** M_i in B_i, based on the framework model, identify the key concepts that these information elements represent. In terms of OO and UML modeling, use the information collected in the previous step to build an object diagram, where instances of classes represent the key concepts (perhaps already identified in the formal framework description) and properties take the values from the message elements:

    $M_i(C_1(E_1, E_2, \ldots), C_2(E_m, E_n, \ldots), \ldots, C_n(E_x, E_y, \ldots))$

    This notation means that each message M_i contains a set of key concepts (classes) – information elements, which decide the meaning of the message.
  - **Collect the key concepts** in a unique set:

    $F_1(C_1, C_2, \ldots, C_n, \ldots, C_x, \ldots, C_z)$

  (NOTE: this is a bottom-up approach. Needs to be re-worked to better reflect the overall top-down approach).
  (NOTE 2: this step corresponds to the process of building conceptual topology of frameworks F1 and F2, which are sets of conceptual neighborhoods [CID52]).
- **Collect more semantic data** about each concept, as expressed by each framework's specifications, in a form of properties and constraints:

  $C_i(p_1, p_2, \ldots, p_m, c_1, c_2, \ldots, c_x)$

  We introduce the notation P_i to denote a property with its accompanying constraints. Therefore we may express the above as follows:

  $C_i(P_1, P_2, \ldots, P_m, c_n, \ldots, c_x)$

  These additional semantic data will probably point to some obvious generalizations, which in turn may lead to reduction of the set of unique concepts.

(NOTE 1: The steps detailed above lead to creation of framework ontologies – or, in the language of [UMM], Lexicons with core components. Similarly, the process described below corresponds to finding a translation between ontologies [OB00] – although, since the ontologies are built from scratch here, the approach to use shared vocabulary may provide useful reduction in complexity (cf. [OB00]). The latter approach is similar to the process described in [ebCDDA] for discovery of domain components and context drivers).
(NOTE 2: the Business Operational View [UMM] model of the frameworks, if available, is a very appropriate source for this kind of information)
(NOTE 3: two concepts $F_1(C_x)$ and $F_2(C_y)$ may in fact represent one real entity – however, due to the different contexts in which they are described they may appear to be non-equal. Such cases will be resolved in the following steps)

- **Generate hypotheses about corresponding concepts** in the other framework:
  - Concepts are likely to correspond if they:
    - have similar properties
    - are similarly classified
    - play similar roles (similar relationships with other concepts, occur in similar contexts)
- **Test each hypothesis:**

Semantic Translation	
Input	Ontologies for each framework, containing the key concepts
Output	Semantic Translation rules, defining the correspondence between the key concepts
Alternative Procedures	
BUSTER	Approximate re-classification (described below)
Subsumption	Check the constraints on the properties, describe the differences in property specifications (such as scale, allowed values, code lists, classification) and check the correctness of classification based on the following criteria: • The **necessary conditions** for concept $F_i(C_x)$ is set of values/ranges of some of its properties that are true for all instances of that concept. Therefore, if a concept C_y doesn't display them, it cannot be classified as C_x. Necessary conditions help to rule out false correspondence hypotheses. • The **sufficient conditions** for concept $F_i(C_x)$ is a set of properties and constraints, when met automatically determine the concept classification. Sufficient conditions help us to identify the concepts that surely correspond because they show all sufficient conditions. Example: "TV-set" meets sufficient conditions for being a "house appliance". However, it fails to meet the necessary conditions for a "cleaning house appliance".
Anchor-PROMPT	
Cupid	
MOMIS	
Ontomorph	
Upper-level ontology labeling	*(using terms from upper-level ontology to label the concepts, and then prepare translation formulas based on the formal subsumption algorithms)*

Approximate re-classification
If the above steps result in well-defined rules of correspondence for most cases of the observed concept occurrence, the hypothesis can be considered basically true. It is probably not feasible to strive for exact solution in 100% cases – we may allow certain exceptions. There are several ways to determine the level of proximity:

- **Rough classification:** the concept definition can be treated as having its upper and lower bounds. The upper bound (the most precise) is necessary conditions, and the lower bound (the most general) is the sufficient conditions. We may declare that $F_1(C_x)$ _ $F_2(C_y)$ even when necessary conditions are not met, but sufficient ones are.
- **Probabilistic classification:** we can determine (based on e.g. available pre-classified data sets) the significance of each property on the result of classification, and so calculate the probability of entity belonging to a specific class.
- **Fuzzy classification:** for each property we define a fuzzy rule, which describes the level of similarity of the tested property. Then, the best match is defined when maximum number of rules gives positive results.

- **Other hypotheses**: if the hypothesis cannot be proven with a sufficient degree of certainty, other hypotheses need to be formulated and tested.
- **Possible difficulties** that may arise:
  - There is **no corresponding** concept: may be there are too many unknown properties to determine the corresponding concept in F_2, because in the context of F_1 they were irrelevant. In this case, the information required to find $F_2(M_x(C_y))$ needs to be supplied from elsewhere, based on properties of the real entities that

$F_1(M_i(C_j))$ and $F_2(M_x(C_y))$ refer to - we need to provide more semantics about the concepts than what is found in the framework specifications (usually from a human expert).

- There are **many corresponding** concepts, depending on which property we choose: we could arbitrarily choose the one that plays the most vital role from the business point of view – and choose which properties decide that an instance of a concept in F1 could be classified as an instance of corresponding concept in F2:
  $F_1(C_x(P_i))$ _ $F_2(C_y(P_j))$
  See also the section above on probabilistic classification.
- The **conflicts in property** constraints cannot be easily resolved. This case calls for help from the domain expert.
- **Describe the rules and exceptions** (if any), and in what contexts they occur.

(NOTE: there are three ways to address this problem, according to [OB00]:

- *Create a single global ontology, which will include concepts from both frameworks. Not feasible for even moderately complex cases.*
- *Create mappings between concepts in ontologies (this is the approach suggested above, although [OB00] warns again that it leads to very complex mappings)*
- *Using shared vocabulary, re-build the ontologies from scratch – the result will be somewhat automatically aligned. Then, prepare the translation rules, which should be now much simpler.)*

4. Syntax translation (to be completed)

- **Data element mapping**
  (NOTE: describe how the external formats can be mapped to internal representation …)

- **Message format mapping**
  (NOTE: describe how the message well-formedness rules can be satisfied. This may involve proactive "asking" for more information in order to satisfy the demands of a given message format…)

- **Message packaging mapping**
  (NOTE: describe how the message packaging [encoding, charset, MIME, etc] can be aligned)

- **Transport protocol mapping**
  - **Align packaging and transport** protocols, based on the specifications in each framework.
- *(to be continued…)*

8 Annex 2 – Example Architecture of ECIMF-compliant Toolkit

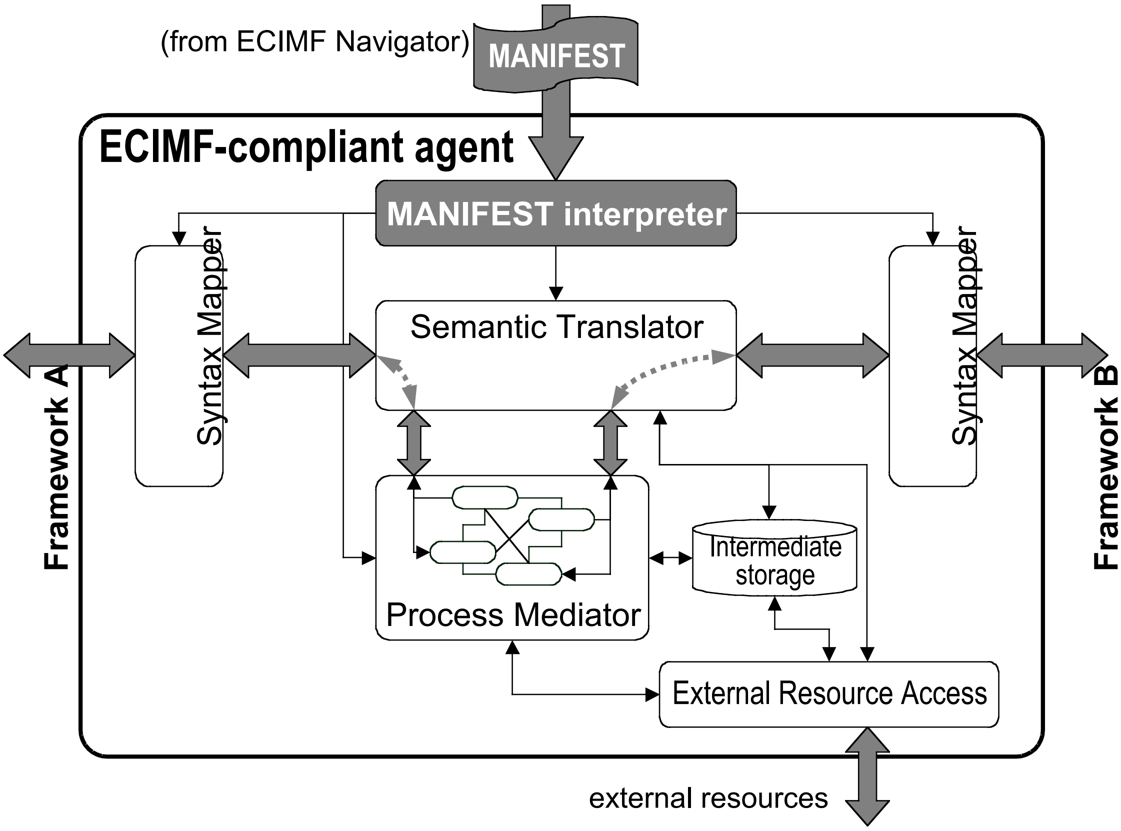


Figure 17 Example of ECIT (ECIMF-compliant agent) facilitating message exchange.

Figure above presents a block diagram of an ECIMF-compliant integration agent. The data flow (represented by thick gray arrows) goes first through the low-level data format adapters (named "Syntax Mappers"), then proceeds to the "Semantic Translators" module, and finally is controlled by the "Process Mediator". The "MANIFEST Interpreter", which uses the information provided in the "MANIFEST" specification prepared in the ECIMF Navigator, configures all these building blocks.

It is important to note that in this model, the ECIMF-compliant agent operates not only on the currently arrived data in the current message, but also uses some historical data stored in the intermediate storage, as well as the data available from external resources.

8.1.1 Syntax Mapper

The syntax Mapper is responsible for translating the message format and transport protocol to/from the internal model representation, which is then used by other modules. This could involve e.g. translating from EDI to XML, and then building an XML Document Object Model (DOM) tree as a representation of the incoming message. Further processing in the Semantic Translation module processes that internal model representation.

8.1.2 Semantic Translator

This module is responsible for changing the information model according to the translation rules, so that the information contained in the original message is understandable for the other party according to its (different) data model and meaning. This module operates only on the internal representation of the data.

8.1.3 Process Mediator

This module aligns the conversational patterns of each of the frameworks. It should be noted that this might require working not only with the currently received data in the message, but also with some historical data in the context of the same conversation. Also, there may be a need for using a given piece of information later during the same conversation, as specified by the differing message formats. For these reasons, the process mediator needs to use an intermediate storage, in which the data related to the context of current conversation may be kept.

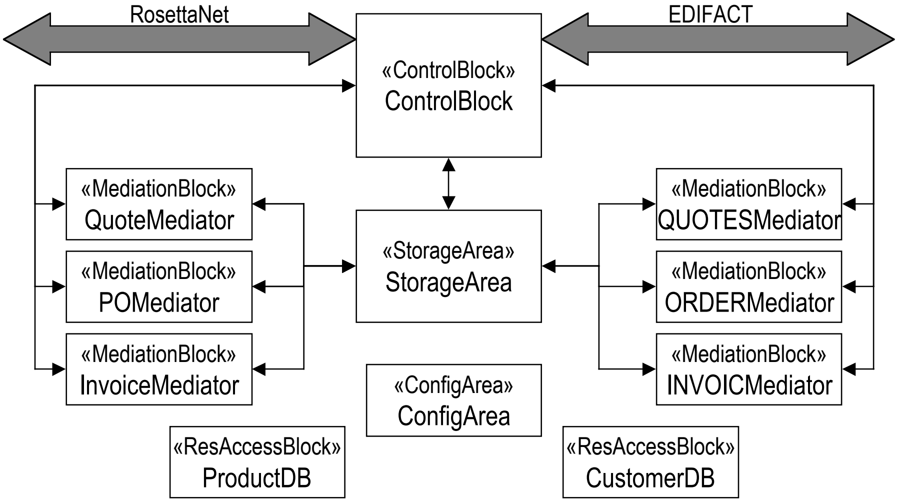


Figure 18 Process Mediator model.

Process Mediator needs to collect all information available from the input messages, complement them with information from other resources (e.g. external directories, configuration parameters), and generate appropriate output messages, which contain necessary information in order to complete given transactions, according to the target framework specification and within its timing constraints.

9 Annex 3 – MULECO: Multilingual Upper-Level Electronic Commerce Ontology

9.1 Editor’s note

The following material included in this Annex has been created as a draft proposal for a separate CEN/ISSS project. This material is far from being complete, and – since the project hasn’t been started due to the lack of resources – cannot be completed at this stage in this forum. However, in the opinion of EC Workshop members it provides a good starting point for anyone wanting to continue this work, and because it discusses the issues of semantic interoperability and the use of ontologies in e-commerce, it has been decided that it should be incorporated verbatim into the final ECIMF CWA as an informal Annex.

ECIMF Project Group gratefully acknowledges the efforts of Mr. Martin Bryan as the primary author of this initiative and editor of the following material.

9.2 What the project hopes to achieve

This CEN/ISSS Electronic Commerce Workshop project will research the most efficient means of developing a multilingual upper-level ontology for describing and identifying the relationships between electronic commerce applications and the ontologies used to describe them. In particular it will investigate how information related to business processes can be integrated with existing techniques for classifying businesses, their products and services.

There are many existing and proposed "electronic commerce ontologies". The vast majority have been defined monolingually, or in at most three or four languages, often from the same language group. The problem is that different trading partners tend to use different ontologies, and tend to prefer ontologies developed in their native language or in a "neutral" language, which is often English. It is, therefore, difficult to identify points of overlap between ontologies, and it is also difficult for people to find relevant terms in ontologies using their native language.

Figure 19 The relationship of MULECO to eCommerce Applications

The aim of MULECO is to develop a mechanism that will allow existing ontologies to identify their inter-relationships by identifying the relationships between themselves and a set of terms defined in a multilingual ontology that has been designed specifically to allow people to find terms using their native language. We realise that it is not possible, or desirable, to create and maintain a multilingual ontology that covers all terms used in all business applications in all European languages. What is needed is a way of classifying entries at the upper-most levels of existing ontologies in a form that takes account of the sort of terms used by people when they are trying to locate the term(s) they wish to use. To do this we need to extend existing business classification schemes to take account of things like business processes, variant names within different user communities, exclusion properties (e.g. no peanuts), etc. Such extensions need to be based on a well documented model that is based on properly researched linguistic characteristics, such as that provided by the Expert Advisory Group in Language Engineering Standards in The EAGLES Guidelines for Lexical Semantic Standards provided in Chapter 6 of *EAGLES LE3-4244: Preliminary Recommendations on Lexical Semantic Encoding -- Final Report* (http://www.ilc.pi.cnr.it/EAGLES96/EAGLESLE.PDF).

The MULECO project will develop an upper-level ontology, expressed as an extended network of industry descriptors, commercial terms and business roles, that will be recorded in a way that allows each entry to be addressed from other ontologies and applications by means of a Uniform Resource Identifier or an XML Path/Query.

The upper-level ontology will take as its start point existing standardized industry and process classification schemes, such as the International Standard for Industrial Classification (ISIC) used as the basis for the NACE classification of European business. The project will take note of the work being done by the IST CLAMOUR project to formally define such classification schemes. In particular it will extend currently used techniques for data classification, based on hierarchical classification of terms into broader and narrower meanings, by allowing for more complex relationships, in particular those relating to the relationships of wholes and parts which are vital to the mapping of the relationships between business processes. By defining a set of business relevant relationships between terms the project will allow classification hierarchies to become a controlled network of related words that forms an ontology rather than a classification scheme.

The ontology will be expressed in a language that provides the following functionality not currently found in electronic commerce ontologies based on languages such as RDF, OIL, KIF, etc, which the members of the CEN/ISSS Electronic Commerce Workshop feel are required to model different kinds of relationships between multilingual electronic commerce ontologies:

1. The ability to uniquely identify the domain (e.g. industry sector) in which each term is employed
2. The ability to formally record the meaning of the term within a particular domain
3. The ability to identify other domains in which the same meaning applies
4. The ability to record alternative terms that have the same meaning within the original domain
5. The ability to identify alternative terms used for the same meaning in other domains
6. The ability to identify an exactly equivalent term used in a different language
7. The ability to identify a nearly equivalent term used in a different language
8. The ability to identify terms that form a part of an object defined by a term
9. The ability to identify wholes that a term forms a part of
10. The ability to identify an opposite term or property (e.g. water-resistant/water-soluble)
11. The ability to record relationships between terms or properties
12. The ability to identify opposite relationships (e.g. isMother/isChild)
13. The ability to declare properties that record measurements
14. The ability to declare properties that record times
15. The ability to associate terms with specific points in process chains

Monolingual ontologies that are linked to the multilingual ontology will be able to make use of equivalences expressed in the multilingual ontology to extend their search potential. This will allow companies that have developed electronic commerce applications for a single country/language to extend their applications to other European countries and beyond without having to change their underlying data dictionaries. With the forthcoming extension of the European Single Market into Eastern Europe and the Mediterranean there will be an increasing need for tools that allow the creation and maintenance of complex multilingual business ontologies of the type to be developed by this project. The project will evaluate the problems associated with developing multilingual ontologies, methodologies and techniques for overcoming them and the advantages to be gained from their use.

This project will incorporate and build on the concepts currently being developed to introduce monolingual ontologies into the Semantic Web. It will introduce such concepts into electronic commerce applications that are aimed at improving the flow of information between businesses within different language communities. At present most of the development work on the Semantic Web is postulated on the basis of using English language terms to identify the relationship between web resources and ontologies. Existing tools for applying the World Wide Web Consortium (W3C)'s Resource Description Framework (RDF) to the identification of related resources are generally postulated on the manual indexation of resources. Business applications require that this work be automated so that resource relationships can be identified automatically in a timely manner as part of business processes, without any human intervention. To be able to do this in a multilingual environment requires the use of a new generation of methodologies and tools. The project will seek to develop methodologies and tools for the creation and maintenance of multilingual ontologies, and for the querying of such ontologies.

The project hopes to:

1. Develop a methodology for expressing a general-purpose ontology for describing the full gamut of electronic commerce applications in multiple languages
2. Develop an open source tool to support the development and maintenance of a multilingual upper-level ontology
3. Populate a multilingual ontology with Internet-addressable terms for describing electronic commerce applications and services, and the relationships between them
4. Identify a set of existing electronic commerce ontologies and associate them with relevant terms in the multilingual upper-level ontology.
5. Input draft specifications into the European and international standardization process.

The results of the project will be reviewed by members of the CEN/ISSS Electronic Commerce Workshop and other relevant standardization organizations.

9.3 Existing Techniques

The following techniques have been studied as possible bases for MULECO:

- The EAGLES Guidelines
- Techniques for the Definition of Ontologies
    - IEEE Standard Upper-level Ontology (SUO)
    - DAML+OIL
- A Thesaurus Interchange Format in RDF
- XML Representation of ISO 13250 Topic Maps
- Unified Modeling Language (UML)

- The Object-Role Modeling (ORM)
- The Common Warehouse Metamodel (CWM) Business Nomenclature Package
- ISO 11179: Specification and Standardization of Data Elements
- ISO DIS 16642:Terminological Markup Framework (TMF)
- ISO 704: Principles and methods of terminology
- The International Standard for Industrial Classification (ISIC)

9.4 The EAGLES Guidelines

The EAGLES project was concerned with Natural Language Processing (NLP). As such it had a very wide theme, and needed to cater for the large number of circumstances in which text is used. Many of its features were concerned with word disambiguation in different contexts that are not directly applicable to the more limited applications for which business semantics are required. This section only discusses those features of the EAGLES Guidelines that are directly relevant for the description of business semantics.

The EAGLES Guidelines for Lexical Semantic Standards provided in Chapter 6 of *EAGLES LE3-4244: Preliminary Recommendations on Lexical Semantic Encoding -- Final Report* (http://www.ilc.pi.cnr.it/EAGLES96/EAGLESLE.PDF) points out that:

> “Hierarchical networks [describing hyperonym/hyponym (broader/narrower term) relationships] are very powerful structures because classifications at the top can be inherited to large numbers of word meanings that are directly or indirectly related to these top levels.”

and

> “to achieve consistency in encoding hyponymy relations, the best approach is to build the hierarchy top down starting from a limited set of tops or unique beginners … Having an overview of the classes, even at a very high level, makes it possible to more systematically check the possible classes. Furthermore, a systematized top level makes it easier to compare and merge different ontologies.”

Business semantics will need someone to develop a top level hierarchy suitable for business uses if they are to be able to interoperate.

As is pointed out in the EAGLES Guidelines, many thesauri cluster words that are related in an unstructured way. For example, the standardized medical thesaurus MESH contains the following entries related to transportation:

```
Transportation
... Aviation
... ... Aircraft
... ... ... Air Ambulances
... ... Space Flight
... ... ... Extravehicular Activity
... ... ... Spacecraft
```

The terms `Space Flight` and `Extravehicular Activity` do not represent subclasses of transportation vehicles but are, rather, types of activities related to certain vehicles. Because of this, MESH can only be used to globally extract words that are related; it cannot be used to make inferences such as: all the things that can be used to transport people, goods, etc.

Ontologies of business semantics need to be structured in such a way that they can be used for making inferences.

Words can have different meanings in different contexts. A term that has more than one meaning is said to exhibit polysemy. Words that share the same meaning within a particular context are synonyms. Synonyms should be able to replace each other in stated contexts. If their replacement is not always possible they are referred to as near-synonyms. Near-synonyms have meanings that partially overlap each other. Terms that share the same parent hyperonym but do not overlap in meaning are known as co-homonyms. It is important that ontologies of business semantics be able to make these distinctions within the relationships they record.

Word-sense disambiguation is an important subtask for Information Retrieval, Information Extraction or Machine Translation. One of the key factors in disambiguation is the identification of the domain with which the relevant text is concerned. If you have identified the domains in which each meaning of a term applies you can disambiguate meanings by utilizing information relating to the domains of discourse within a resource.

While hyperonym/homonym relationships work for nouns they are not so useful for other parts of speech, which are generally harder to disambiguate. For most business related classification schemes, however, verbs and other parts of speech are of relatively low importance in identifying meaning. (Verbs identify relationships or actions: they can be useful to identify the role played by particular agents on particular objects. Roles can be classified to create thematic roles. Adjectives are used to describe properties of nouns, e.g. brown gloves. Adverbs, prepositions, conjunctions, etc, are not widely used in electronic business messages. Of key importance to business, however, are terms used for the quantification of measurements and for defining time.)

Many lexicons permit multiple hyperonyms (broader terms) to be associated with a homonym (narrower term). Three types of hyperonym have been identified within the EAGLES project: exclusive, conjunctive and non-exclusive. For exclusive hyperonyms one of a choice of meanings must be determined by context. Conjunctive hyperonyms allow more than one meaning to be associated with a given context. If either multiple meanings or a single meaning can apply in a given context the hyperonym is deemed to be non-exclusive.

The EAGLES-based EuroWordNet distinguishes between Entities, Concepts, Events and States. Each of these is further divided, with up to 5 levels of subdivision. A typical EuroWordNet entry has the form:

```
[ -ORTHOGRAPHY : horse
  -WORD-SENSE-ID : horse_1
  -BASE-TYPE-INFO : [ BASE-TYPE: ANIMAL
                      LX-RELATION: LX-HYPONYM]
                    [ BASE-TYPE: OBJECT
                      LX-RELATION: LX-HYPONYM]
  SYNONYMS : Equus_caballus_1
  HYPERONYMS : [HYP-TYPE: conjunctive
                HYP-ID: animal_1]
               [HYP-TYPE: conjunctive
                HYP-ID: equid_1]
               [HYP-TYPE: non-exclusive
                HYP-ID: pet_1]
               [HYP-TYPE: non-exclusive
                HYP-ID: draught_animal_1]
  HYPONYMS : [HYP-TYPE: disjunctive
              HYP-ID: mare_1]
```

```
            [HYP-TYPE: disjunctive
              HYP-ID: stallion_1]]
```

Meronymy is defined as a lexical part-whole relationship between elements. A good example is provided by human body parts. "Finger" is a meronym of "hand" which is a meronym of "arm" which is a meronym of "body". The "inverse relation" is called holonymy. “Body" is the holonym of "arm" which is the holonym of "hand" which is the holonym of "finger". The co-meronymy relationship is one between lexical items defining sister parts (arm, leg, head are co-meronyms of body). Meronymy is different from taxonomy because it does not classify elements by class. That is to say, the hierarchical structuring of meronymy does not originate in a hierarchy of classes (toes, fingers, heads, legs, etc, are not hierarchically related).

Not all meronyms are related to a single holonym. For example, "nail" is more general than its holonym "toes" as it can also be part of a finger as well. Cruse[8] introduced the notions of super-meronym ("nail" is a super-meronym of "toes") and hypo-holonym ("toes" is a hypo-holonym of "nail") to allow for this.

The EAGLES paper recommends that "any lexical semantic standard should record a simple binary relation of antonymy where possible between [opposite] word senses". For example, "north" is the antonym of "south", and vice versa.

All of the above techniques can usefully be applied to multilingual business ontologies.

The on-going work, within the ISLE project for the development of International Standards for Language Engineering (http://www.ilc.pi.cnr.it/EAGLES96/isle/), on a Multilingual ISLE Lexical Entry (MILE) will extend the EAGLES Guidelines to cover the relationships between entries in different languages.

9.5 Techniques for the Definition of Ontologies

An ontology is a particular system of categories that provides a certain vision of the world. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships (e.g. lower-level terms meet the criteria set for higher-level terms). An ontology is the general framework within which catalogues, taxonomies, terminologies, etc, may be organized.

The key ingredients that make up an ontology are a vocabulary of terms and a precise specification of what those terms mean. But ontologies also analyse the fundamental categories of *objects, their current state,* and whether they form a *part* or the *whole* of something else, as well as the relations between parts and the whole and their laws of dependence.

Recently ontologists have started to explore the potential of process- and task-based ontologies. Rather than trying to describe all the characteristics of a particular universe or business domain, these more limited, time-aware, ontologies seek to provide information that is relevant for the management of a particular process or the completion of a specified task. One advantage of taking this approach is that it is much easier to develop intelligent agents that can make inferences based on such specialized domain knowledge than it is to create general-purpose inference engines of the type typically employed in AI systems. At this stage the question of how to identify the relationship between ontologies developed for specific processes and tasks has not been explored. Because it includes facilities for identifying the processes used by a particular business domain, however, MULECO should make it easier for intelligent agents, as well as human users, to identify ontologies that are relevant to their particular processes and tasks.

[8] Cruse, D. A. 1986. *Lexical Semantics*. Cambridge University Press

A formal ontology is the result of combining the intuitive, informal method of classical ontology analysis with the formal, mathematical method of modern symbolic logic. Over the years a wide range of formal ontologies have been proposed. To make it possible for ontologies to exchange data a number of "knowledge representation languages" have been developed, including KIF, Ontolingua, SNePS, HOL and Conceptual Graphs. Of these the most influential seems to have been the Knowledge Interchange Format (KIF). The basis for the semantics of KIF is a conceptualization of the world in terms of objects and relations among those objects. There are nine types of terms in KIF -- individual variables, constants, character references, character strings, character blocks, functional terms, list terms, quotations, and logical terms.

9.5.1 IEEE Standard Upper-level Ontology (SUO)

KIF, which is in the process of being published as a US standard by ANSI (see http://logic.stanford.edu/kif/dpans.html), has been chosen by IEEE as the basis for a Standard Upper-level Ontology (SUO). This upper ontology is limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas. As well as very high level constructs such Independent Entity and Relative Entity SUO will cover such things as Agents, Persons and Organizations, using KIF definitions of the form:

```
(subclass-of Agent Object)
(subclass-of Person Agent)
(subclass-of Organization Agent)
(subclass-of Publisher Organization)
(subclass-of University Organization)
(disjoint Person Organization)
(subclass-of LegalObligation InstitutionalObligation)
```

and constructs for basic business functions, such as:

```
(subclass-of Quantity SpatialForm)
(subclass-of Weight Quantity)
(subclass-of Arrangement Schema)
(subclass-of Number Arrangement)
(subclass-of Set Arrangement)
```

SUO will also define instances of particular relationships, using formulations such as:

```
(instance-of hasAnnotation BinaryRelation)
(nth-domain hasAnnotation 1 Object)
(nth-domain hasAnnotation 2 TextObject)
```

and

```
(instance-of subProcess BinaryRelation)
(nth-domain subProcess 1 Process)
(nth-domain subProcess 2 Process)
```

Definitions can be assigned to SUO concepts using documentation statement of the form:

```
(documentation Agent "An active animate entity that voluntarily initiates an
action.")

(documentation Arrangement "Mathematical structures that do not have spatial
dimensions: numbers, sets, lists, algebras, grammars, and the data structure of
computer science. Arrangement includes the subclasses whose names are derived
from _taxis_, the Greek word for "arrangement", including taxonomies and syntax.
```

All the syntactic forms in natural languages, programming languages, and versions of symbolic logic are included under Arrangement.")

As was the case with the all-encompassing lexical approach proposed by EAGLES, the proposed Standard Upper-level Ontology is designed to cover all knowledge, and therefore starts with concepts that are at much too high a level for the integration of business processes. It would be more correct to call it the Standard Top-level Ontology as it is designed to encompass all ontologies, rather than provide an upper level for a set of ontologies that cover specific areas, of the type proposed for the Multilingual Upper-Level Electronic Commerce Language.

Note: MULECO is not designed to integrate all existing ontologies, or to provide a meta-schema for describing ontologies. It is strictly limited to providing a means of identifying the relationships between existing ontologies by providing them with a set of addressable shared terms that they can link their top-levels to.

9.5.2 DAML+OIL

The Ontology Inference Language (OIL) that has been adopted as part of the DARPA Agent Markup Language (DAML) is an application of the W3C Resource Description Framework (RDF). DAML+OIL (http://www.daml.org/2001/03/reference.html) divides the world up into objects, which are elements of DAML classes, and datatype values, i.e., values that come from XML Schema datatypes, like the integer 4.

In DAML+OIL an ontology is recorded using a set of definitions that define classes, subclasses, properties that connect classes and individual instances. Classes have names, descriptive documentation, statements of which class it creates a subclass of, and one or more constraining facets. Classes are allowed to have multiple superclasses, which are deemed to be conjunctive unless specifically defined as being disjoint. DAML+OIL properties are divided into two sorts, those that relate objects to other objects and those that relate objects to datatype values. The former belong to daml:ObjectProperty while the latter belong to daml:DatatypeProperty. Properties are defined as having ranges of permitted values. Multiple ranges can be applied to a property but then the value of the property must satisfy all range statements (they are conjunctive rather than disjoint, with only the intersection of all the statements being valid). Properties, but not their values, can be defined as being the inverse of each other

DAML Class definitions can be defined in multiple statements, as the following parts of a March 2001 DAML Class definition example illustrate:

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#age"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
<daml:Class rdf:about="#Person">
  <rdfs:comment>Every person is a man or a woman</rdfs:comment>
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Man"/>
    <daml:Class rdf:about="#Woman"/>
  </daml:disjointUnionOf>
</daml:Class>
...
<daml:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <daml:Restriction daml:maxCardinality="1">
      <daml:onProperty rdf:resource="#hasSpouse"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
...
<daml:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <daml:Restriction daml:maxCardinalityQ="1">
      <daml:onProperty rdf:resource="#hasOccupation"/>
      <daml:hasClassQ rdf:resource="#FullTimeOccupation"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

DAML classes are a subset of the RDF Schema (RDFS) Class construct. The rdfs:SubclassOf element that forms its first level contents is extended by the use of the daml:Restriction definition. Whilst this leads to a more detailed definition of DAML classes it does mean that there is a confusion between classes of the type used for defining schemas in RDF and the types of categorization used to define an ontology.[9]

An instance of the DAML Class shown above might take the form:

```
<Person rdf:ID="Peter">
  <rdfs:comment>
  Peter is an instance of Person. Peter has shoesize 9.5 and age 46
  </rdfs:comment>
  <shoesize>9.5</shoesize>
  <age><xsd:integer rdf:value="46"></age>
</Person>
```

Each DAML ontology can have associated with it metadata that identifies what the ontology is about, the version of DAML being used, and other information relevant to the management of the ontology. Ontologies can import part or all of another ontology.

A typical DAML+OIL header takes the form:

[9] The classes used in programming are typically additive in nature, properties at a lower level being added to those at higher levels. Categories in ontologies, in contrast, are restrictive in nature, the properties at one level distinguishing subsets of the properties applicable at a higher level.

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
 xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
 xmlns:dex ="http://www.daml.org/2001/03/daml+oil-ex#"
 xmlns:exd ="http://www.daml.org/2001/03/daml+oil-ex-dt#"
 xmlns     ="http://www.daml.org/2001/03/daml+oil-ex#" >
<daml:Ontology rdf:about="">
 <daml:versionInfo>$Id: daml+oil-ex.daml,v 1.9 2001/05/03 16:38:38
                   mdean Exp $</daml:versionInfo>
  <rdfs:comment>
    An example ontology, with data types taken from XML Schema
  </rdfs:comment>
  <daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
</daml:Ontology>
```

9.5.3 A Thesaurus Interchange Format in RDF

The IST LIMBER project has prepared a paper defining *A Thesaurus Interchange Format in RDF* (http://www.limber.rl.ac.uk/External/SW_conf_thes_paper.htm) which is based on the techniques defined in ISO 2788:1986 *Documentation--Guidelines for the establishment and development of monolingual thesauri* (2nd ed., 1986) and ISO 5964:1985 *Documentation--Guidelines for the establishment and development of multilingual thesauri* (1985). The model defines RDF classes for Thesaurus Object, Concept, Top Concept, Term, Scope Note and Scope Note Type. Concepts can have the following properties: Classification Code, In Language Of, Has Scope Note, Is Indicated By, Concept Relation and Concept Equivalence. The Is Indicated By property has subproperties of Preferred Term and Used For. The Concept Relation property has subproperties of Broader Concept, Narrower Concept, Top of Hierarchy, Top Concept and Is Related To. The Concept Equivalence property has subproperties of Exact Equivalent, Inexact Equivalent, Partial Equivalent and One-to-many Equivalent. Scope Notes have the property of In Language Of and Has Type Of, where permitted types are General, Hierarchy, Translation, Editor and History. The following example illustrates how these terms are applied:

```
<rdf:Description rdf:ID="EN620">
 <rdf:type rdf:resource=
    “http://www.data-archive.ac.uk/Limber/RDF/thesaurus#Concept”/>
 <thes:ClassificationCode>EN620</thes:ClassificationCode>
 <thes:inLanguageOf rdf:resource=
            "http://www.data-archive.ac.uk/Limber/ISO639#en"/>
 <thes:PreferredTerm>
  <rdf:Description>
   <rdf:type rdf:resource=
    "http://www.data-archive.ac.uk/Limber/RDF/thesaurus#Term"/>
    <rdf:value>friends</rdf:value>
  </rdf:Description>
 </thes:PreferredTerm>
 <thes:UsedFor>
  <rdf:Description>
   <rdf:type rdf:resource=
      "http://www.data-archive.ac.uk/Limber/RDF/thesaurus#Term"/>
   <rdf:value>mates</rdf:value>
  </rdf:Description>
 </thes:UsedFor>
 <thes:hasScopeNote>
  <rdf:Description>
   <rdf:type rdf:resource=
    "http://www.data-archive.ac.uk/Limber/RDF/thesaurus#ScopeNote"/>
   <thes:inLanguageOf rdf:resource=
     "http://www.data-archive.ac.uk/Limber/ISO639#fr"/>
   <thes:hasTypeOf rdf:resource=
     "http://www.data-archive.ac.uk/Limber/RDF/thesaurus#Regular"/>
    <rdf:value>To be used only for Platonic relationships.
    </rdf:value>
 </rdf:Description>
 </thes:hasScopeNote>
 <thes:BroaderConcept rdf:resource="#EN983"/>
 <thes:NarrowerConcept rdf:resource="#EN329"/>
 <thes:RelatedConcept rdf:resource="#EN243"/>
 <thes:TopConcept rdf:resource="#EN345"/>
 <thes:ExactEquivalent rdf:resource="#FR620"/>
</rdf:Description>
```

9.5.4 XML Representation of ISO 13250 Topic Maps

The XML Topic Maps (XTM) specification provides a model and grammar for representing the structure of information resources used to define *topics*, and the associations (relationships) between topics. Names, resources, and relationships are said to be *characteristics* of topics. Topics can have their characteristics defined within *scopes* that limit the contexts within which the names and resources are regarded as meaningful. One or more interrelated documents employing this grammar is called a “topic map”.

A minimal topic, consisting of a base name and a single resource identified as an occurrence of the topic, could be defined as:

```
  <topic id="hamlet">
    <instanceOf><topicRef xlink:href="#play"/></instanceOf>
    <baseName>
      <baseNameString>Hamlet, Prince of Denmark</baseNameString>
    </baseName>
    <occurrence>
      <instanceOf>
        <topicRef xlink:href="#plain-text-format"/>
      </instanceOf>
      <resourceRef
       xlink:href="ftp://www.gutenberg.org/pub/1ws2610.txt"/>
    </occurrence>
  </topic>
```

An association representing the relationship between Shakespeare and the play *Hamlet* might look like this:

```
  <association>
    <instanceOf><topicRef xlink:href="#written-by"/></instanceOf>
    <member>
      <roleSpec><topicRef xlink:href="#author"/></roleSpec>
      <topicRef xlink:href="#shakespeare"/>
    </member>
    <member>
      <roleSpec><topicRef xlink:href="#work"/></roleSpec>
      <topicRef xlink:href="#hamlet"/>
    </member>
  </association>
```

Within topic maps, scopes establish the contexts in which a name or an occurrence is assigned to a given topic, and the context in which topics are related through associations. Any topics having the same base name in the same scope implicitly refer to the same subject and therefore should be merged.

XTM, unlike the underlying ISO standard, privileges two types of association: class-instance, and superclass-subclass. It fails, however, to follow the ISO standard in permitting the assignment of user-defined facets to provide multi-dimensional views of topic maps.

9.5.5 The Unified Modeling Language (UML)

UML is the main technique used for modelling business processes. It forms the basis of the UN/CEFACT Modeling Methodology (UMM), Version 10 of which can be found at http://www.gefeg.com/tmwg/n090r10.htm. UMM forms the basis for modelling business processes within the ebXML/ebTWG initiative to establish a new generation of business messaging services that is compatible with XML.

An overview of the ebXML process is provided in the following UML diagram:

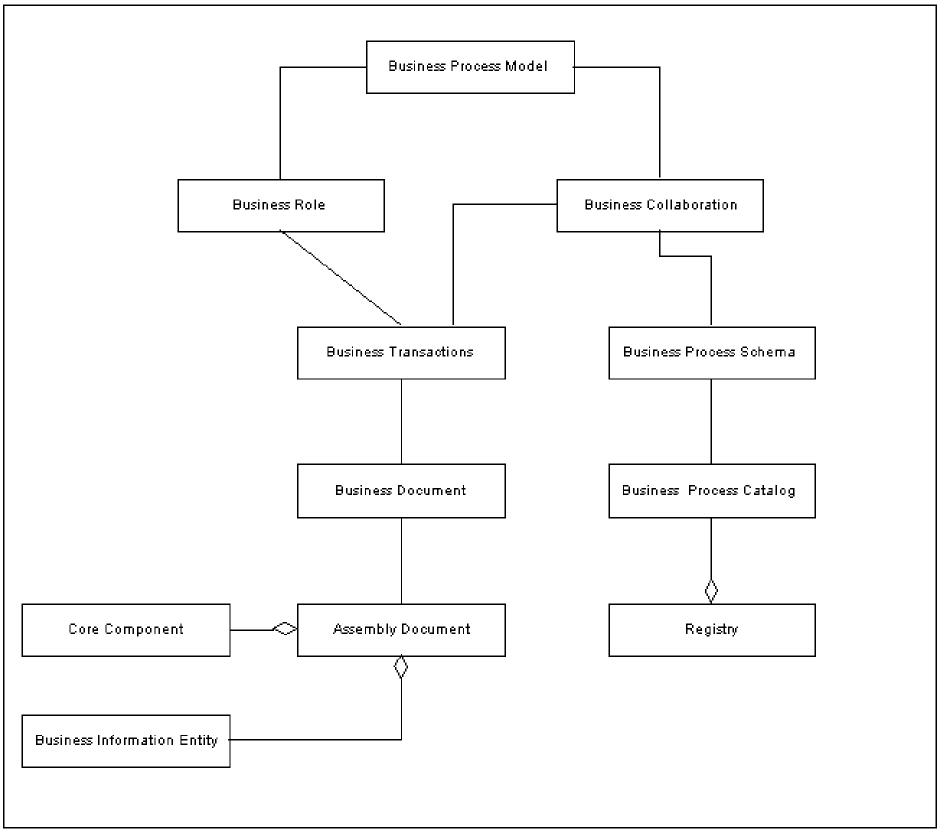
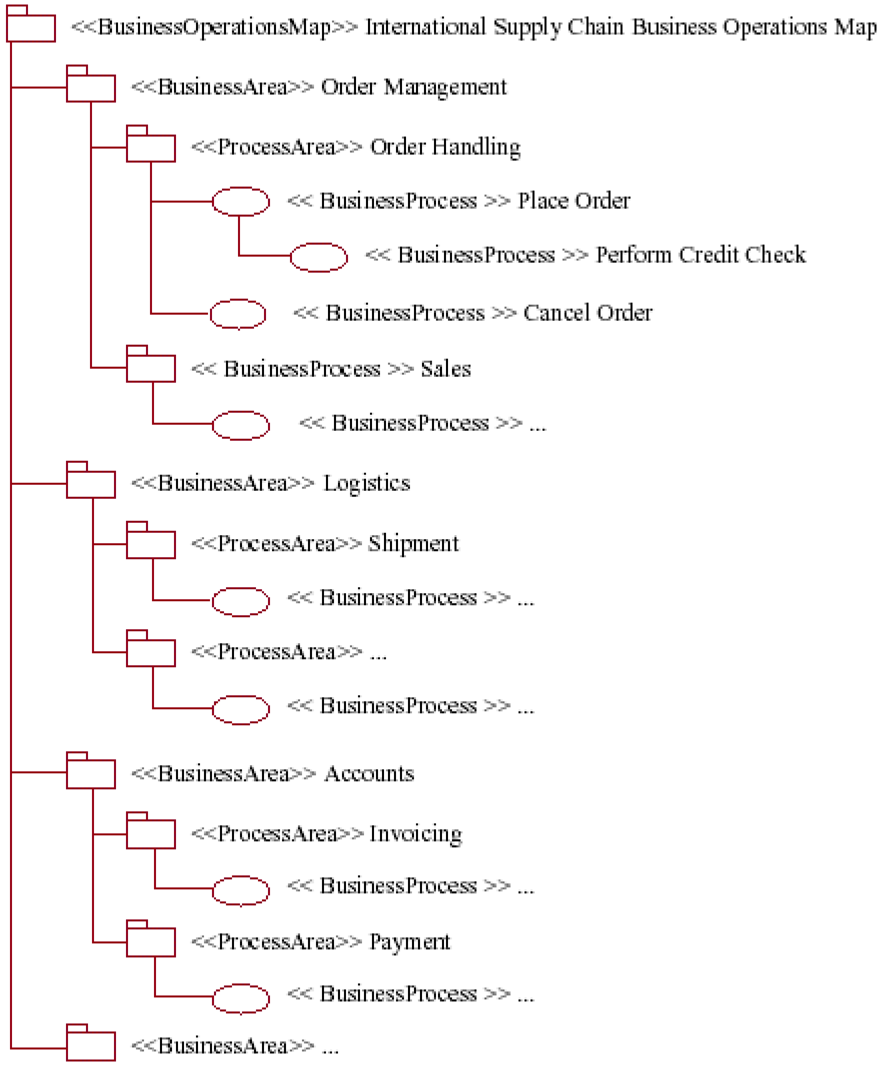


Figure 20 Core ebXML concepts

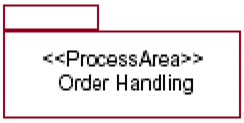
UML has problems with the correct definition of sequence, an important feature of business process management, but does allow choice and multiple triggering inputs to be clearly identified.

From a linguistic point of view a UML model can be thought of as a set of *entities* (things that have names which are nouns) that are linked by *relationships* (lines whose names are verbs). Entities have *properties* (whose values provide adjectives that qualify the nouns) and relationships have *qualifications* (whose values provide adverbs that qualify the verbs).

A UMM Business Operations Map (BOM), which forms the basis for ebXML business models, has the following typical representation:


<<BusinessOperationsMap>> International Supply Chain Business Operations Map
<<BusinessArea>> Order Management
<<ProcessArea>> Order Handling
<< BusinessProcess >> Place Order
<< BusinessProcess >> Perform Credit Check
<< BusinessProcess >> Cancel Order
<< BusinessProcess >> Sales
<< BusinessProcess >> ...
<<BusinessArea>> Logistics
<<ProcessArea>> Shipment
<< BusinessProcess >> ...
<<ProcessArea>> ...
<< BusinessProcess >> ...
<<BusinessArea>> Accounts
<<ProcessArea>> Invoicing
<< BusinessProcess >> ...
<<ProcessArea>> Payment
<< BusinessProcess >> ...
<<BusinessArea>> ...

where the following example definitions are used to define the Order Handling Process Area and Place Order Business Process:



CategorySchema:	UN/CEFACT's Ontology
Category:	UN.CEFACT.SupplyChain.OrderManagement.OrderHandling
Objective:	Facilitate key activities relative to order handling between buyer/seller for products and services.
Scope:	Limited to order handling activities, excluding sales.
Business Justification:	Facilitation of order handling activities.

Precondition:	A contract or a framework agreement exists
BeginsWhen:	The customer recognises a need for a products and contacts a seller.
Definition:	The customer recognises a need for a product and places an Order under an established contact or a framework agreement. The supplier receives order and responds.
EndsWhen:	Ends when order conditions have been met or order is cancelled.
Exceptions:	• Seller is not able to deliver the goods. • Seller is not allowed to sell the goods (trade embargoes, government regulations, etc.) • …
Postcondition:	The customer has received positive acknowledgement of product scheduled delivery.

The Centre for User-oriented IT Design (CID) at the Swedish Royal Institute of Technology (KTH) have developed a technique for generalizing UML models to provide Unified Language Modeling (ULM) that allows formal models to be expressed in terms that are easily understood by businesses. The following diagrams summarize this technique:

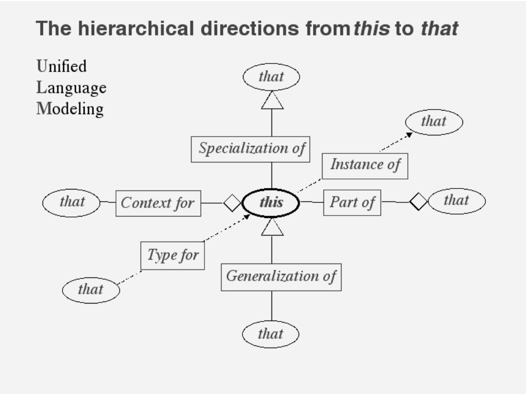


Figure 21 The basic principles for Unified Language Modeling

Using this technique you can understand that:

- The concept called car represents kind of vehicle
- The concept called vehicle is an abstraction of the concept called car
- The concept called wheel forms a part of a car
- A car has one or more wheels
- A specific car (:car) is an instance of the car concept
- A specific wheel (:wheel) is an instance of the wheel concept
- A specific wheel is a part of a specific car
- A specific car is a kind of vehicle

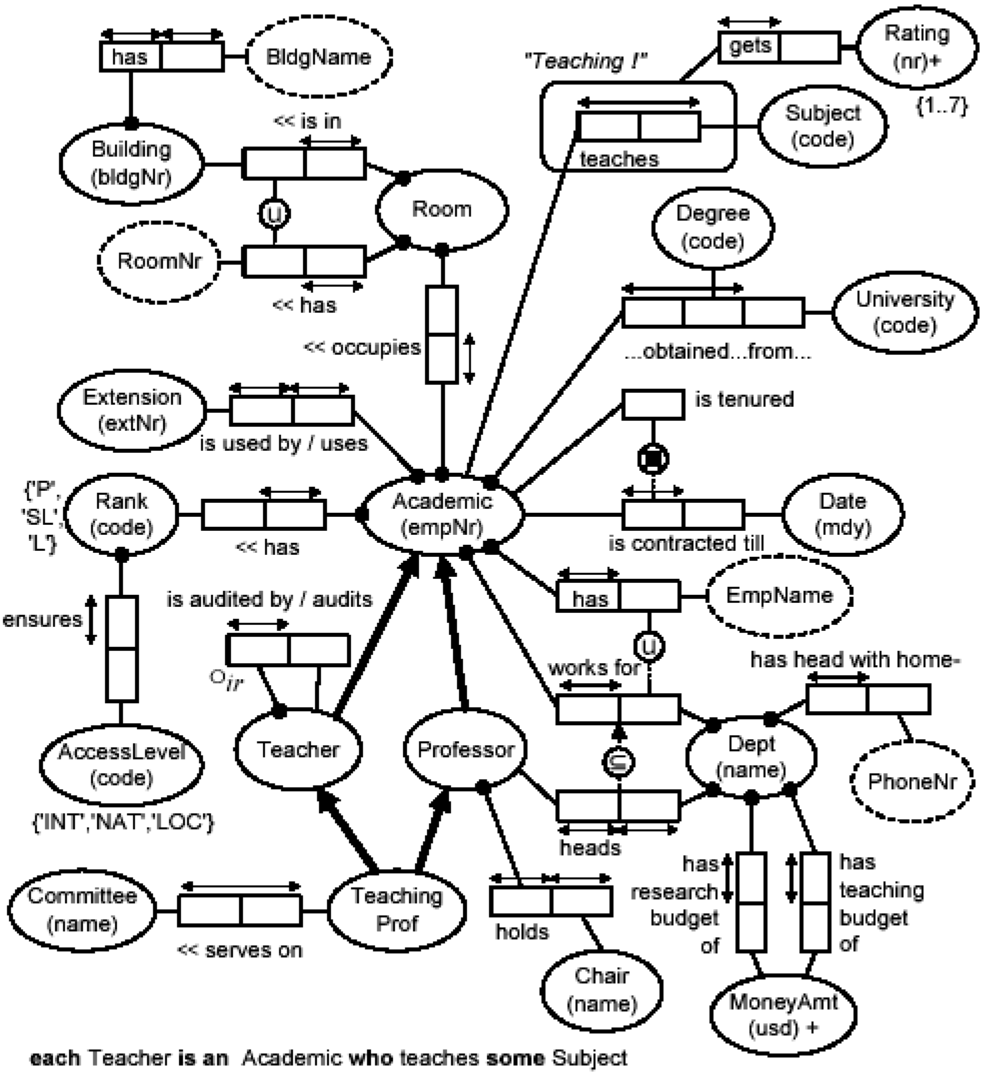
9.5.6 The Object-Role Modeling (ORM)

An alternative to UML is the Object-Role Modeling (ORM) promoted by Microsoft. ORM's *conceptual schema design procedure* (CSDP) focuses on the analysis and design of data. The conceptual schema specifies the information structure of the application: the *types of fact* that are of interest; *constraints* on these and *derivation rules* for deriving certain facts from others. The stages involved are:

1. Transform familiar information examples into elementary facts, and apply quality checks
2. Draw the fact types, and apply a population check
3. Check for entity types that should be combined, and note any arithmetic derivations
4. Add uniqueness constraints, and check arity of fact types
5. Add mandatory role constraints, and check for logical derivations
6. Add value, set comparison and subtyping constraints
7. Add other constraints and perform final checks

Object-Role Modeling is so-called because it views the world in terms of objects playing roles. Facts are assertions that objects play roles. An *n*-ary fact has *n* roles. It is not necessary that the roles be played by different objects.

A typical ORM diagram has the form:



each Teacher **is an** Academic **who** teaches **some** Subject
each Professor **is an** Academic **who** has Rank 'P'
each TeachingProf **is both a** Teacher **and a** Professor

* Dept employs academics of Rank in Quantity **iff** Quantity =
count each Academic **who** has Rank **and** works for Dept

* **define** Extension provides AccessLevel **as**
Extension is used by **an** Academic **who** has **a** Rank **that** ensures AccessLevel

Figure 22 ORM diagram.

9.5.7 The Common Warehouse Metamodel (CWM) Business Nomenclature Package

The following diagram summarizes the parts of the Open Management Group's Common Warehouse Metamodel (OMG CWM) Business Nomenclature Package:

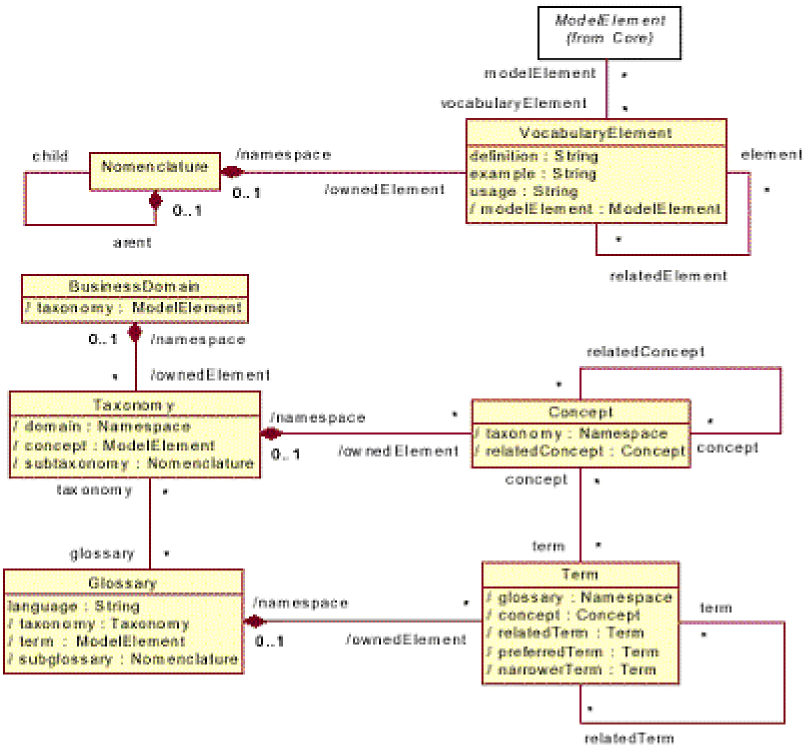


Figure 23 OMG's CWM core concepts.

The OMG model considers Taxonomies to consist of a number of Concepts, which may or may not have Related Concepts. A Taxonomy may be related to a Glossary, which contains one or more Term, which may have a Preferred Term, a number of Related Terms and one or more Narrower Terms. Terms can be related to Concepts in a Taxonomy.

9.5.8 ISO 11179: Specification and Standardization of Data Elements

As noted in ISO DTR 20943, *Procedures for achieving metadata registry (MDR) content consistency*, the two types of abstraction of most interest to data element development are *specialization/generalization* and *decomposition/aggregation*.

Specialization/generalization is a relationship between two classes, where all items in one (subclass) are also in the other (superclass). Decomposition/aggregation relates an item to its parts. Decomposition may be described as "x is a part of y," or the *part-of* relationship. The reverse, aggregation, shows that y may be composed of x among other items.

The ISO/IEC 11179-3 metamodel does not provide for linking of data elements. A registration authority recording such data elements, however, might choose to extend the model to link data elements based on their layers of abstraction, including generalization to specialization, and other relationships. Linkages can occur in both vertical relationships (e.g., from general to more specific) and horizontal relationships (e.g., with equivalent layers of specialization). They can also be linked according to other relationships (e.g., data elements that are always used together). Vertical relationships are those where a specialized data element that has been registered for a particular purpose is related to a generalized data element that is intended for a general purpose. Horizontal relationships are those where data elements with different names have equivalent definitions that represent the same layer of specialization, with equivalent data value domains.

When using a top-down approach to ISO 11179 data element registration, small amounts of data are often added to a registry in groups or rather than as individual data elements. When a classified group of data elements is to be added to the registry, the analyst might choose to identify the conceptual domains that are relevant to the group, consider their value meanings, and work down to data elements.

Figure below shows the Order of Registering Components for Top Down Registration of a Data Element defined in ISO DTR 20943.

1 Conceptual Domain (CD)

- CD Context
- CD Name
- CD Definition
- CD Identifier
- CD Registration Status
- CD Administrative Status

2 Value Meaning (VM)

- VM Description
- VM Begin Date
- VM End Date
- VM Identifier

3 Data Element Concept (DEC)

- DEC Context
- DEC Name
- DEC Definition
- Object Class
- Object Class Qualifier
- Property
- Property Qualifier
- DEC Identifier
- DEC Registration Status
- DEC Administrative Status

4 Value Domain (VD)

- VD Context
- VD Name
- VD Definition/Description
- VD Identifier
- Datatype
- Minimum Characters
- Maximum Characters
- Unit of Measure
- Precision
- VD Origin
- VD Explanatory Comment
- Permissible Values (PV)
- PV Begin Date
- PV End Date
- Representation Class
- Representation Class Qualifier
- VD Registration Status
- VD Administrative Status

5 Data Element (DE) Definition and Name

- DE Context
- DE Definition
- DE Name
- Registration Authority Identifier
- DE Identifier:Version Identifier
- DE Example
- DE Origin
- DE Comment
- DE Registration Status
- DE Administrative Status

Order of Registering Components for Top Down Registration of a Data Element

9.5.9 Terminological Markup Framework (TMF)

ISO TC37/SC3 has published a Draft International Standard (DIS 16642, December 2001) that defines a Terminology Markup Framework. This International Standard specifies a model that has been designed for the purpose of providing guidance on the basic principles for representing terminological data, as well as for describing specific terminological markup languages. It is based on the principles laid down in ISO 704:2000, *Principles and methods of terminology* and provides an annex that shows how the framework can be applied to ISO 12620, *Computer applications in terminology - Data categories*.

The relationship of the component parts of the TMF model is shown in Figure below:

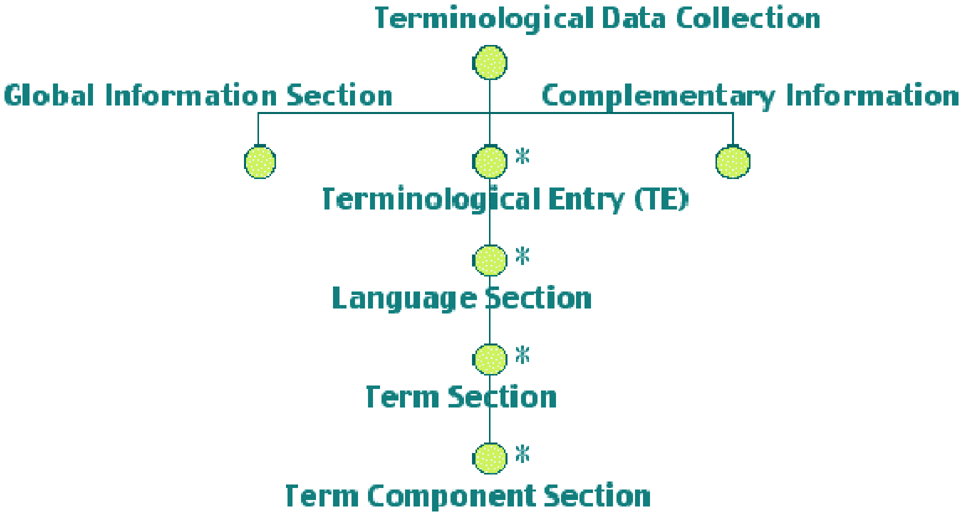

Terminological Data Collection
Global Information Section
Complementary Information
*
Terminological Entry (TE)
*
Language Section
*
Term Section
*
Term Component Section

Figure 24 Terminological Markup Framework (TMF) Metamodel

The component parts of this diagram are:

- **TDC (Terminological Data Collection):**
  Top level *container* for all information contained in a terminology system. Generally used as a *container* for other containers in the system - may contain descriptive information such as, in XML, the validating schema that would be used.
- **GIS (Global Information Section):**
  Information that applies to all elements represented in a file, as opposed to information that may pertain to some, but not to all components of the file.
  Usually contains, for example, the title of the (XML) file, the institution or individual originating the file, address information, copyright information, update information, etc.
- **CI (Complementary Information):**
  Usually contains, for example, textual bibliographical or administrative information residing in or external to the file, static or dynamic graphic images, video, audio, or virtually any other kind of binary data. Might also include references to other terminological resources or contextual links to related text corpora or to ontologies.
- **TE (Terminological Entry):**

Information that pertains to a single concept. Usually contains, for example, the terms assigned to a concept, descriptive information pertinent to a concept, and administrative information concerning the concept. Can contain one or more language sections.

- **LS (Language Section):**
  Contains all the terminological sections (TS) for a terminological entry that are used in a given language, as well as information such as definitions, contexts, etc. associated with that language or the terms in that language.
- **TS (Term Section):**
  Information about terms, including definitions, contexts, etc, associated with the term.
- **TCS (Term Component Section):**
  Information about morphemic elements, words, or contiguous strings from which a polynomial term is formed. In languages such as French or Spanish it is important to be able to include information such as gender for the individual words used in constructing a multiword term because this information is necessary when using the term in texts.

The standard also defines a Generic Mapping Tool (GMT) using XML as the description language. The meta-model can be represented by means of a generic element <struct> (for *structure*) which can recursively express the embedding of the various representation levels of a terminological data collection. The role of each structural node within the meta-model is identified by means of a *type* attribute associated with the <struct> element, i.e., TDC, GIS, TE, CI, LS, TS, TCS. Basic information units associated with a structural skeleton are represented using the <feat> (for *feature*) element. Compound information units are represented using the <brack> (for *bracket*) element, which can itself contain a <feat> element followed by any combination of <feat> elements and <brack> elements. The content model of the <feat> element can contain annotations expressed by means of an <annot> (for *annotation*) element. A *type* attribute can be used to reference an ISO 12620 data category or an equivalent user-defined data category.

An entry in the GMT format might take the form:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<tmf>
 <struct type="TE">
  <feat type="entry identifier">ID67</feat>
  <feat type="subject field">manufacturing</feat>
  <feat type="definition">A value between 0 and 1 used in ...</feat>
    <struct type="LS">
     <feat type="language identifier">en</feat>
      <struct type="TS">
       <feat type="term">alpha smoothing factor</feat>
       <feat type="term type">fullForm</feat>
      </struct>
    </struct>
    <struct type="LS">
     <feat type="language identifier">hu</feat>
       <struct type="TS">
         <feat type="term">Alfa simítási tényezõ</feat>
       </struct>
    </struct>
 </struct>
</tmf>
```

An annex in the standard shows how the meta-model can be used to repersent an ISO 12200 *MARTIF-compatible format with specified constraints (MSC)* using the TMF Typed Element Style. Using this method the above entry takes the format:

```
<?xml version='1.0'?>
```

```
<!DOCTYPE martif SYSTEM "./MSCcoreStructureDTD-v-1-0.DTD.TXT">
<martif type='MSC' xml:lang='en' >
  <martifHeader>
    <fileDesc>
      <sourceDesc>
       <p>from an Oracle corporation termBase</p>
      </sourceDesc>
    </fileDesc>
    <encodingDesc>
     <p type='DCSName'>MSCdefaultXCS-v-1-0.XML</p>
    </encodingDesc>
  </martifHeader>
  <text>
   <body>
    <termEntry id='eid-Oracle-67'>
     <descrip type='subjectField'>manufacturing</descrip>
     <descrip type='definition'>A value between 0 and 1 used in ...
     </descrip>
     <langSet xml:ang='en'>
       <tig>
        <term tid='tid-Oracle-67-en1'>alpha smoothing factor</term>
        <termNote type='termType'>fullForm</termNote>
      </tig>
     </langSet>
     <langSet xml:lang='hu'>
      <tig>
       <term tid='tid-Oracle-67-hu1'>Alfa simítási tényezõ</term>
      </tig>
     </langSet>
    </termEntry>
   </body> </text>
</martif>
```

9.5.10 ISO 704: Principles and methods of terminology

The “Principles of term formation” defined in this standard includes the following:

- A term should be attributed to a single concept.
- Terms should be “transparent”: a term is considered to be transparent when the concept it designates can be inferred, at least partially, without a definition.
- The terminology of any subject field should provide a coherent terminological system corresponding to the concept system.
- Terms should adhere to familiar, established patterns of meaning within a language community.
- Terms should be as neutral as possible: avoid negative connotations.
- Terms should be as concise as possible.
- Term formations that allow derivatives should be favoured.
- Terms should conform to the mrophological norms of the language.
- Native language expressions should be given preference over direct loans.

9.5.11 The International Standard for Industrial Classification (ISIC)

ISIC Version 3.0 (ISIC3) is the primary scheme used by governments throughout the world to classify business activity. It forms the basis of the Euopean NACE classification of EU economic activity. ISIC uses the following top-level hierarchy:

- A - Agriculture, hunting and forestry

- B - Fishing
- C - Mining and quarrying
- D - Manufacturing
- E - Electricity, gas and water supply
- F - Construction
- G - Wholesale and retail trade; repair of motor vehicles, motorcycles and personal and household goods
- H - Hotels and restaurants
- I - Transport, storage and communications
- J - Financial intermediation
- K - Real estate, renting and business activities
- L - Public administration and defence; compulsory social security
- M - Education
- N - Health and social work
- O - Other community, social and personal service activities
- P - Private households with employed persons
- Q - Extra-territorial organizations and bodies

Each of these subdivisions is further subdivided. For example, the Manufacturing subdivision is further subdivided into:

- 15 - Manufacture of food products and beverages
- 16 - Manufacture of tobacco products
- 17 - Manufacture of textiles
- 18 - Manufacture of wearing apparel; dressing and dyeing of fur
- 19 - Tanning and dressing of leather; manufacture of luggage, handbags, saddlery, harness and footwear
- 20 - Manufacture of wood and of products of wood and cork, except furniture; manufacture of articles of straw and plaiting materials
- 21 - Manufacture of paper and paper products
- 22 - Publishing, printing and reproduction of recorded media
- 23 - Manufacture of coke, refined petroleum products and nuclear fuel
- 24 - Manufacture of chemicals and chemical products
- 25 - Manufacture of rubber and plastics products
- 26 - Manufacture of other non-metallic mineral products
- 27 - Manufacture of basic metals
- 28 - Manufacture of fabricated metal products, except machinery and equipment
- 29 - Manufacture of machinery and equipment
- 30 - Manufacture of office, accounting and computing machinery
- 31 - Manufacture of electrical machinery and apparatus
- 32 - Manufacture of radio, television and communication equipment and apparatus
- 33 - Manufacture of medical, precision and optical instruments, watches and clocks
- 34 - Manufacture of motor vehicles, trailers and semi-trailers
- 35 - Manufacture of other transport equipment
- 36 - Manufacture of furniture
- 37 – Recycling

These upper levels of the ISIC scheme show the typical problems that occur when you try to group together subjects into a single hierarchy. For example, the topmost level A shows that activities related to land use other than those associated with property have been grouped together, but the name applied to them does not reflect the reason for this grouping, but instead is simply the sum of the lower-level activities that make up the group.

Similar problems occur in the relationships between different levels. For example, the Manufacture of food products and beverages is part of the Manufacturing set of activities (D15) and not in any way associated with the production of the raw materials used therein, which generally comes under the heading Agriculture. Similarly the Manufacture of basic metals is not associated with the Mining and quarrying needed to obtain the raw materials for the processes.

These problems could be overcome by adopting a polyhierarchical system, which allowed the fact that the inputs from one chain came from the lower levels of a second chain, but to date most industrial classification schemes rely on single levels of nested classes.

It is unlikely that people will use the terms used in the ISIC headings as the basis for asking questions about classification schemes. It is, therefore, necessary to consider what terms users are likely to use to identify each of these terms. A more natural set of simple-to-understand top-level headings might include:

- Food
- Energy
- Raw materials
- Manufactured Goods
- Retailing
- Financial Services
- Transport
- Recreation
- Personal Services
- Property
- Civil Engineering
- Education
- Health and Social Services
- Public Administration
- Non-profit organizations

It should be noted that the ISIC listing is only available in three languages, English, French and Spanish. Translations into other languages would be needed to provide a truly multilingual classification scheme.

9.6 Proposed Approach

The ontology representation language should be expressed in XML so that individual components of it can be referenced as component parts of either a Unique Resource Indicator (URI), XML Path definition or XML Query.

The underlying structure of the XML should be based on the concepts described in the EAGLES framework, but with alternative forms of element names based on typical business renditions of technical terms (e.g. BroaderTerm in place of Hypernym). The terms to be adopted form EAGLES, and their equivalent business terms are shown in the following table:

Linguistic Terminology	Ontological Terminology	Business Terminology
Phrase	Concept	Term/Name
Hypernym	Superclass	Broader Term
Holonym	Subclass	Narrower Term
Synonym	Synonym	Alternative Term
Near-Synonym		Partially Matches / Includes
		No Equivalent*
Holonym		Forms Part Of
Meronym		Has Part / Has Process
Antonym		Opposite
	Restriction	Constraint Excludes
	Association	UsedBy RelatedTo ReverseOf
Measurement		MeasurementsRequired MeasurementType PermittedUnit
Time		TimesRequired TimeType

* Indicates that a particular language has no matching term

Entries should be provided with metadata which is defined by reference to existing sources of information or by use of standardized metadata descriptors. Each term must be assigned to at least one subject domain, ideally by linking it to a standardized domain identified within ISIC.

These terms can be used to create the following XML DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD for defining the Multilingual Upper-Level Electronic Commerce Ontology
(MULECO)
© Martin Bryan of The SGML Centre (www.sgml.u-net.com) 25th January 2002 -->
<!--Entity used to indicate that attribute value should be expressed as a valid
URL (or as a pointer to an ID assigned in the ontology).-->
<!ENTITY % URL "CDATA">
<!ELEMENT Ontology (Term|TimeType|MeasurementType|PermittedUnit)+>
<!ATTLIST Ontology
  Region %URL; #IMPLIED
  Industry %URL; #IMPLIED
  Process %URL; #IMPLIED
>
<!ELEMENT Term (SubjectDomain+, Definition+, Name+, NoEquivalent*,
 (AlternativeTerm | Synonym)*, (PartiallyMatches | Includes |
 NearSynonym)*, (BroaderTerm | Hypernym)*, (NarrowerTerm | Hyponym)*,
 (FormsPartOf | Holonym)*, (HasPart | HasProcess | Meronym)*,
 (Opposite | Antonym)*, (Restriction | Constraint | Excludes)*,
 (UsedBy | RelatedTo | ReverseOf)*, (MeasurementsRequired |
 TimesRequired)*)>
```

```
<!ATTLIST Term
 ID ID #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
>
<!--Pointer to subject domains that this term is used within.
 (Subject domains may be defined in a separate onotology.)-->
<!ELEMENT SubjectDomain (Label*)>
<!ATTLIST SubjectDomain
 xlink:href  %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Formal definition of term. Must be at least one. If more than one
 the values of their xml:lang attributes must differ.-->
<!ELEMENT Definition (#PCDATA | html:p)*>
<!ATTLIST Definition
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #IMPLIED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #IMPLIED
>
<!--Name by which term is known within specified SubjectDomains-->
<!ELEMENT Name (#PCDATA)>
<!ATTLIST Name
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #IMPLIED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #IMPLIED
>
<!--Identifies languages for which there is no equivalent term-->
<!ELEMENT NoEquivalent EMPTY>
<!ATTLIST NoEquivalent
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #IMPLIED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #IMPLIED
>
<!--Alternative name by which term is known in an identified domain
 (must share definition used for name exactly, otherwise it is a
 NearSynonym). (Technically these are referred to as Synonyms, but
 for convenience of use the AlternativeTerm element is defined as an
 equivalent.)-->
<!ELEMENT AlternativeTerm (#PCDATA)>
<!ATTLIST AlternativeTerm
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!ELEMENT Synonym (#PCDATA)>
<!ATTLIST Synonym
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
```

```
<!--Name applied in different language (or domain) that is near to
 current term but not exactly synonymous (i.e. does not share exactly
 the same definition). If the term overlaps with the related term
 then it is said to Partially Match it. If it is a subset of the term
 (without overlapping into other areas) then the parent term Includes
 the near synonym.-->
<!ELEMENT PartiallyMatches (#PCDATA)>
<!ATTLIST PartiallyMatches
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!ELEMENT Includes (#PCDATA)>
<!ATTLIST Includes
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!ELEMENT NearSynonym (#PCDATA)>
<!ATTLIST NearSynonym
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Pointer to elements used to define terms whose broader meaning includes this
term. (Technically these are referred to as Hypernyms, but for convenience of
use the BroaderTerm alternative is a defined equivalent.)-->
<!ELEMENT BroaderTerm (Label*)>
<!ATTLIST BroaderTerm
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->-->
<!ELEMENT Hypernym (Label*)>
<!ATTLIST Hypernym
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
 >
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!--Pointer to elements used to define terms whose narrower meaning is included
in this term. (Technically these are referred to as Hyponyms, but for
convenience of use the NarrowerTerm alternative is a defined equivalent.)-->
```

```
<!ELEMENT NarrowerTerm (Label*)>
<!ATTLIST NarrowerTerm
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID .-->
<!ELEMENT Hyponym (Label*)>
<!ATTLIST Hyponym
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->-->
<!--Pointer to elements used to define terms that provide components used to
create the information set defined by term. (Technically these are referred to
as Meronyms, but for convenience of use the HasPart alternative is a defined
equivalent.)-->
<!ELEMENT HasPart (Label*)>
<!ATTLIST HasPart
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!ELEMENT HasProcess (Label*)>
<!ATTLIST HasProcess
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!ELEMENT Meronym (Label*)>
<!ATTLIST Meronym
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
```

```
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!--Pointer to elements used to define terms for which  this term forms a
component of the definition. (Technically these are referred to as Holonyms, but
for convenience of use the FormsPartOf alternative is a defined equivalent.)-->
<!ELEMENT FormsPartOf (Label*)>
<!ATTLIST FormsPartOf
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!ELEMENT Holonym (Label*)>
<!ATTLIST Holonym
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!--Pointer to elements used to define the exact opposite of this term.
(Technically these are referred to as Antonyms, but for convenience of use the
Opposite alternative is a defined equivalent.)-->
<!ELEMENT Opposite (Label*)>
<!ATTLIST Opposite
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element.-->
<!ELEMENT Antonym (Label*)>
<!ATTLIST Antonym
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!--Pointer to documents that define constraints on the use of this term.
(Technically these are referred to as Restrictions, but for convenience of use
the Constraint alternative is a defined equivalent.)-->
<!ELEMENT Constraint (Label*)>
<!ATTLIST Constraint
```

```
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!ELEMENT Restriction (Label*)>
<!ATTLIST Restriction
 xlink:href %URL; #REQUIRED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID.-->
<!--Pointer to components that may not be included in associated products or
processes. (Technically these are also Restrictions, but they are more specific
in application that the more generalized constraints definitions, for which no
formal language has yet been defined.)-->
<!ELEMENT Excludes (Label*)>
<!ATTLIST Excludes
 xlink:href %URL; #REQUIRED
 xml:lang CDATA #IMPLIED
 LabelsID     ID    #IMPLIED
 LabelsRef    IDREF #IMPLIED
 >
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the excluded item as defined in an external
ontology should be entered as the contents of this element. Where more than one
term applies the language of each term must be identified using the xml:lang
attribute. If labels used here are referenced from elsewhere the LabelsID
attribute must be used to name the set. If labels used elsewhere are to be
applied to this element the labelsRef attribute must reference the relevant
LabelsID.-->
<!--Pointer to term that makes use of this term.-->
<!ELEMENT UsedBy (Label*)>
<!ATTLIST UsedBy
 xlink:href %URL; #REQUIRED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the term pointed to in an external ontology
should be entered as the contents of this element. If labels used here are
referenced from elsewhere the LabelsID  attribute must be used to name the set.
If labels used elsewhere are to be applied to this element the labelsRef
attribute must reference the relevant LabelsID. -->
<!--Pointer to term related to this term, together with name that defines the
relationship between this term and the current term.-->
<!ELEMENT RelatedTo (RelationshipName+, RelatedTerm)>
<!ATTLIST RelatedTo
 ID ID #REQUIRED
>
<!--Name used to describe relationship between two or more related terms.-->
<!ELEMENT RelationshipName (#PCDATA)>
```

```
<!ATTLIST RelationshipName
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Where more than one relationship name is defined for a relationship the
language of each name must be identified using the xml:lang attribute. -->
<!--Term to which the current term is to be related.-->
<!ELEMENT RelatedTerm (#PCDATA)>
<!ATTLIST RelatedTerm
 xlink:href %URL; #REQUIRED
 xml:lang CDATA #IMPLIED
>
<!--Element contents can be empty if the URL pointed to is within the same
ontology. Otherwise the name of the excluded item as defined in an external
ontology should be entered as the contents of this element. -->
<!--Relationship (defined as part of another term) that this relationship is the
reverse of.-->
<!ELEMENT ReverseOf (RelationshipName+)>
<!ATTLIST ReverseOf
 RelatesTo  IDREF #REQUIRED
>
<!--Identification of measurements required to fully define term, process or
part-->
<!ELEMENT MeasurementsRequired (MeasurementTypeRef+)>
<!--Pointer to definition of measurement type required.-->
<!ELEMENT MeasurementTypeRef EMPTY>
<!ATTLIST MeasurementTypeRef
 MeasurementType  IDREF #REQUIRED
>
<!--Definition of measurement type consisting of the names by which the
measurement type is referred together with a list of units that are allowed to
be used to specify the measurement.-->
<!ELEMENT MeasurementType (MeasurementName+, PermittedUnitRef+)>
<!ATTLIST MeasurementType
 ID ID #REQUIRED
>
<!--Name used to request measurement.-->
<!ELEMENT MeasurementName (#PCDATA)>
<!ATTLIST MeasurementName
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Where more than one measurement name is defined the language of each name
must be identified using the xml:lang attribute. -->
<!--Pointer to definition of the unit type required.-->
<!ELEMENT PermittedUnitRef EMPTY>
<!ATTLIST PermittedUnitRef
 Unit IDREF #REQUIRED
>
<!-- Definition of unit that can be used for measurement, consisting of the
names by which the unit type is referred together with the string(s) that is
used to qualify numbers defined using this unit (e.g. kg or lbs and ozs).-->
<!ELEMENT PermittedUnit (UnitName+,(UnitIdentifier|PermittedValue)+)>
<!ATTLIST PermittedUnit
```

```
 ID ID #REQUIRED
>
<!--Name used to request measurement.-->
<!ELEMENT UnitName (#PCDATA)>
<!ATTLIST UnitName
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Where more than one name is assigned to the unit the language of each name
must be identified using the xml:lang attribute. -->
<!--String used to identify the use of number as a measurement or time unit.-->
<!ELEMENT UnitIdentifier (#PCDATA)>
<!ATTLIST UnitIdentifier
 Separator CDATA #IMPLIED
 Purpose CDATA #IMPLIED
>
<!--Where two or more sub-units share the same separator they must each be
assigned a different value for the Purpose attribute. Where it has not been
specified, the separator is presumed to be a space or line ending following the
contents of the element.-->
<!--String used to indicate specific value of measurement-->
<!ELEMENT PermittedValue (#PCDATA)>
<!ATTLIST PermittedValue
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Identification of times required to fully define term, process or part-->
<!ELEMENT TimesRequired (TimeTypeRef+)>
<!--Pointer to definition of time type required.-->
<!ELEMENT TimeTypeRef EMPTY>
<!ATTLIST TimeTypeRef
 TimeType IDREF #REQUIRED
>
<!--Definition of time type consisting of the names by which the time type is
referred together with a list of units that are allowed to be used to specify
the time.-->
<!ELEMENT TimeType (TimeName+, PermittedUnitRef+)>
<!ATTLIST TimeType
 ID ID #REQUIRED
>
<!--Name used to request time.-->
<!ELEMENT TimeName (#PCDATA)>
<!ATTLIST TimeName
 xml:lang CDATA #IMPLIED
 RecordedBy CDATA #REQUIRED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #REQUIRED
 SubjectDomain %URL; #IMPLIED
>
<!--Where more than one time name is defined the language of each name must be
identified using the xml:lang attribute. -->
<!--Element used to label a link to a service that is not defined
multilingually.-->
<!ELEMENT Label (#PCDATA)>
```

```
<!ATTLIST Label
 xml:lang CDATA #REQUIRED
 RecordedBy CDATA #IMPLIED
 Organization CDATA #IMPLIED
 WhenRecorded CDATA #IMPLIED
 SubjectDomain %URL; #IMPLIED
>
<!--Element inherited from HTML specification.-->
<!ELEMENT html:p (#PCDATA)>
```

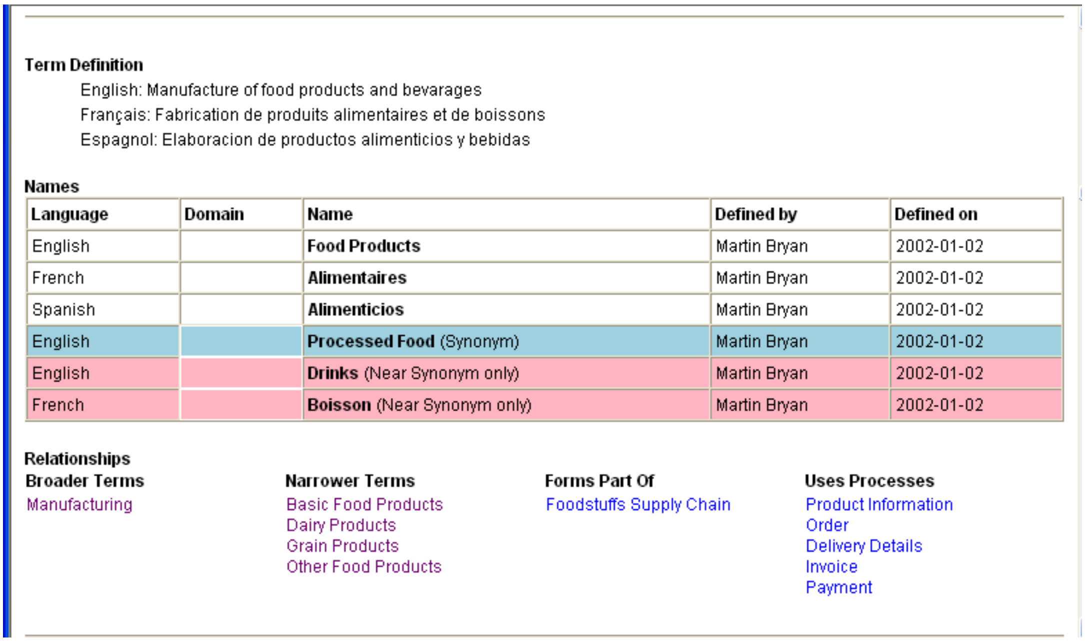
A simplified example of the use of this DTD might have the following form:

```
<Ontology
 Region="http://www.iso.org/ISO639/EU"
 Industry="http://esa.un.org/unsd/registry/ISIC3/">
 ...
<Term ID="ISIC3-D15"
 RecordedBy="Martin Bryan" WhenRecorded="2002-01-02">
  <SubjectDomain xlink:href="#ISIC3-D"/>
  <Definition xml:lang="EN">
   Manufacture of food products and bevarages</Definition>
  <Definition xml:lang="FR">
   Fabrication de produits alimentair&eacute;s et de boissons
  </Definition>
  <Definition xml:lang="ES">
   Elaboraci&oacute;n de productos alimenticios y bebidas
  </Definition>
  <Name xml:lang="EN">Food Products</Name>
  <Name xml:lang="FR">Alimentaires</Name>
  <Name xml:lang="ES">Alimenticios</Name>
  <AlternativeTerm xml:lang="EN"
   RecordedBy="Martin Bryan" WhenRecorded="2002-01-02">
   Processed Food</AlternativeTerm>
  <Includes xml:lang="EN"
   RecordedBy="Martin Bryan" WhenRecorded="2002-01-02">
   Drinks</Includes>
  <Includes xml:lang="FR"
   RecordedBy="Martin Bryan" WhenRecorded="2002-01-02">
   Boisson</Includes>
  <BroaderTerm xlink:href="#ISIC3-D"/>
  <NarrowerTerm xlink:href="#ISIC3-D151"/>
  <NarrowerTerm xlink:href="#ISIC3-D152"/>
  <NarrowerTerm xlink:href="#ISIC3-D153"/>
  <NarrowerTerm xlink:href="#ISIC3-D154"/>
  <FormsPartOf xlink:href="SupplyChain.xml#Food">
    <Label xml:lang="EN">Foodstuffs Supply Chain</Label>
    <Label xml:lang="ES">Comestibles suministro</Label>
    <Label xml:lang="FR">Alimentaire</Label>
  </FormsPartOf>
  <HasProcess xlink:href="BusinessProcesses.xml#ProductInformation">
    <Label xml:lang="EN">Product Information</Label>
    <Label xml:lang="ES">Productos</Label>
    <Label xml:lang="FR">Produits</Label>
  </HasProcess>
  <HasProcess xlink:href="BusinessProcesses.xml#Order">
    <Label xml:lang="EN">Order</Label>
    <Label xml:lang="ES">Orden</Label>
    <Label xml:lang="FR">Ordre</Label>
  </HasProcess>
  <HasProcess xlink:href="BusinessProcesses.xml#Delivery">
    <Label xml:lang="EN">Delivery Details</Label>
    <Label xml:lang="ES">Reparto</Label>
    <Label xml:lang="FR">Livraison</Label>
  </HasProcess>
  <HasProcess xlink:href="BusinessProcesses.xml#Invoice">
    <Label xml:lang="EN">Invoice</Label>
    <Label xml:lang="ES">Factura</Label>
    <Label xml:lang="FR">Facturer</Label>
  </HasProcess>
  <HasProcess xlink:href="BusinessProcesses.xml#Payment">
    <Label xml:lang="EN">Payment</Label>
    <Label xml:lang="ES">Pago</Label>
    <Label xml:lang="FR">Paiement</Label>
  </HasProcess>
</Term>
...
<MeasurementType ID=”weight”>
  <MeasurementName xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Weight</MeasurementName>
  <MeasurementName xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Poid</MeasurementName>
  <MeasurementName xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Peso</MeasurementName>
  <PermittedUnitRef Unit=”kg”/>
  <PermittedUnitRef Unit=”lb”/>
  <PermittedUnitRef Unit=”lbs-and-ozs”/>
  <PermittedUnitRef Unit=”tonnes”/>
  <PermittedUnitRef Unit=”hundredweight”/>
</MeasurementType>
...
<PermittedUnit ID=”kg”>
  <UnitName xml:lang=”EN ES FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>kilo</UnitName>

  <UnitName xml:lang=”EN FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>kilogramme</UnitName>
  <UnitName xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>kilogram</UnitName>
  <UnitIdentifier>kg</UnitIdentifier>
</PermittedUnit>
<PermittedUnit ID=”lbs-and-ozs”>
  <UnitName xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Pounds and ounces</UnitName>
  <UnitIdentifier>lb</UnitIdentifier>
  <UnitIdentifier>oz</UnitIdentifier>
</PermittedUnit>
...
<TimeType ID=”DeliveryDate”>
  <MeasurementName xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Delivery Date</MeasurementName>
  <MeasurementName xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Temp de deliverance</MeasurementName>
  <MeasurementName xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Fecha da reparto</MeasurementName>
  <PermittedUnitRef Unit=”day”/>
  <PermittedUnitRef Unit=”date-and-time”/> >
</MeasurementType>
<PermittedUnit ID=”date-and-time”>
  <UnitName xml:lang=”EN” RecordedBy=”Martin Bryan”
```

```
   WhenRecorded=”2002-01-06”>Date and Time</UnitName>
  <UnitIdentifier Separator=”-” Purpose=”year”/>
  <UnitIdentifier Separator=”-” Purpose=”month”/>
  <UnitIdentifier Separator=”T” Purpose=”date”/>
  <UnitIdentifier Separator=”:” Purpose=”hour”/>
  <UnitIdentifier Separator=”:” Purpose=”minute”/>
</PermittedUnit>
<PermittedUnit ID=”day”>
  <UnitName xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Day and Time</UnitName>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Monday</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Tuesday</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Wednesday</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Thursday</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Friday</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>Saturday</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>lunes</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>martes</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>miércoles</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>jueves</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>viernes</PermittedValue>
  <PermittedValue xml:lang=”ES” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>sábado</PermittedValue>
  <PermittedValue xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>lundi</PermittedValue>
  <PermittedValue xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>mardi</PermittedValue>
  <PermittedValue xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>mercredi</PermittedValue>
  <PermittedValue xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>jeudi</PermittedValue>
  <PermittedValue xml:lang=”FR” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>vendredi</PermittedValue>
  <PermittedValue xml:lang=”EN” RecordedBy=”Martin Bryan”
   WhenRecorded=”2002-01-06”>samedi</PermittedValue>
</PermittedUnit>
...
</Ontology>
```

Using the XSL Transformation Language (XSLT), this file can be converted into an HTML file for display on a web browser in the following format:



Term Definition

English: Manufacture of food products and bevarages
Français: Fabrication de produits alimentaires et de boissons
Espagnol: Elaboracion de productos alimenticios y bebidas

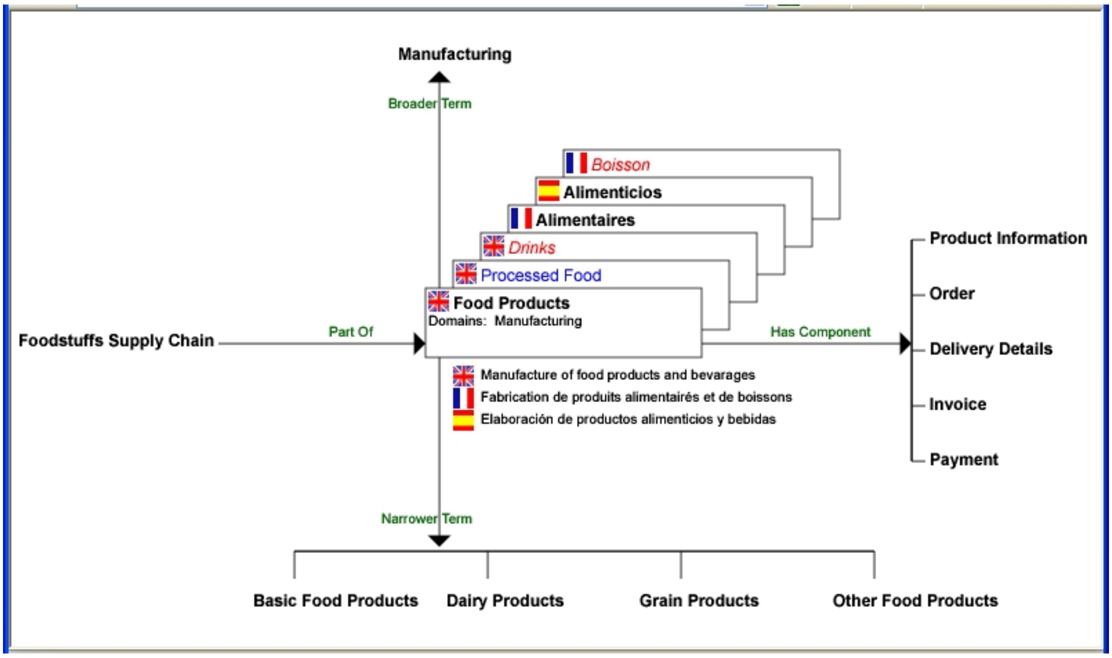
Names

Language	Domain	Name	Defined by	Defined on
English		**Food Products**	Martin Bryan	2002-01-02
French		**Alimentaires**	Martin Bryan	2002-01-02
Spanish		**Alimenticios**	Martin Bryan	2002-01-02
English		**Processed Food** (Synonym)	Martin Bryan	2002-01-02
English		**Drinks** (Near Synonym only)	Martin Bryan	2002-01-02
French		**Boisson** (Near Synonym only)	Martin Bryan	2002-01-02

Relationships

Broader Terms	Narrower Terms	Forms Part Of	Uses Processes
Manufacturing	Basic Food Products Dairy Products Grain Products Other Food Products	Foodstuffs Supply Chain	Product Information Order Delivery Details Invoice Payment

Alternatively the term could be presented graphically as:



9.7 Current Status

MULECO is an on-going project, and as yet no formal set of definitions, or accompanying DTD/Schema, has been agreed. Areas of ongoing study include those currently being undertaken by European research projects such as MILES, CLAMOUR and OntoWeb, and by international e-commerce initiatives such ebXML/ebTWG, related to:

- Formal languages for describing ontologies
- Formal languages for describing multilingual word sets
- Formal models for the maintaining industrial classification schemes
- Formal languages for modelling business processes
- Techniques for the creation and maintenance of process-based ontologies

If you would like to take part in the project please contact our website at http://www.cenorm.be/isss/Workshop/ec/New_Projects.htm#MULECO

Martin Bryan
The SGML Centre
28th January 2002

Annex A: Pictoral representation of MULECO Schema

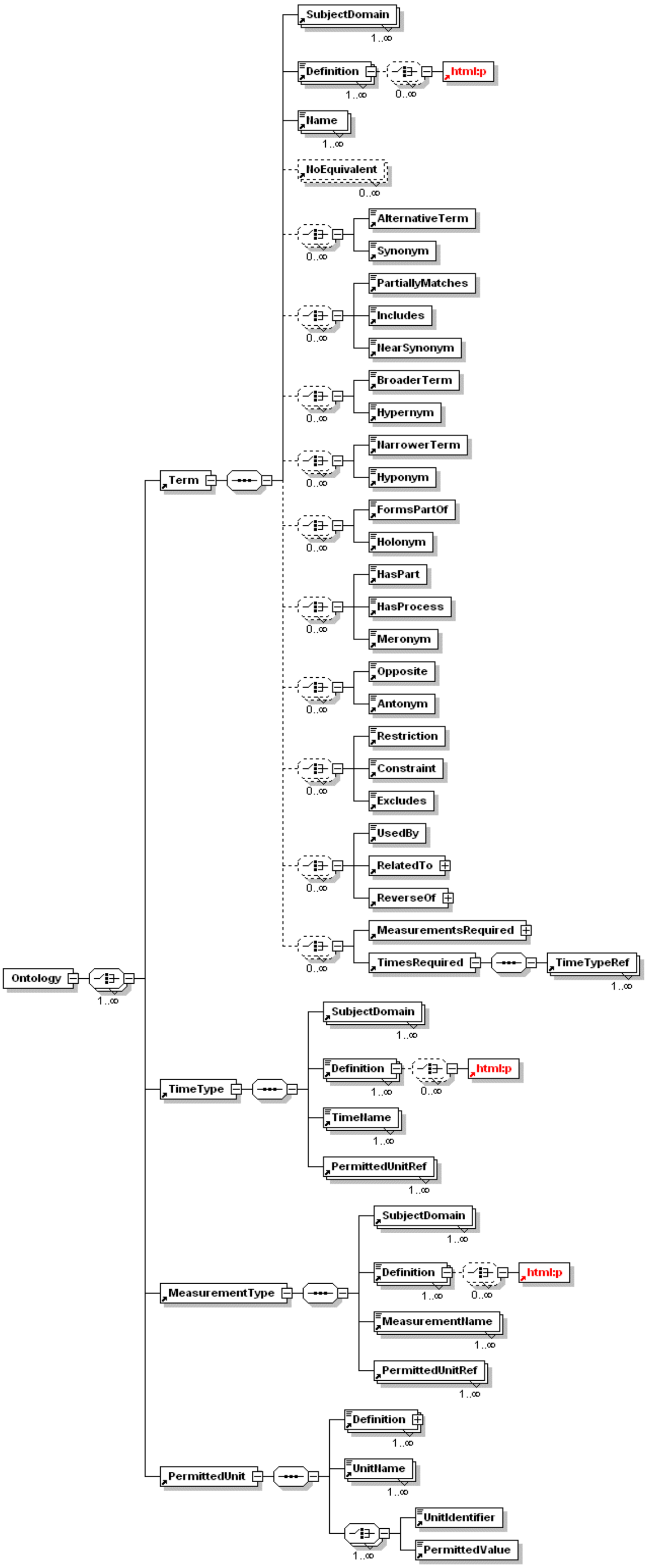


Figure 25 MULECO Schema