TripCom

Triple Space Communication

FP6 – 027324

Deliverable

D7.3

Relationships among Ontology Modules within EDIFACT and with External Ontologies

David de Francisco Marcos, Guillermo López Reyes and douglas foxvog

October 10, 2008

Executive Summary

This deliverable presents the relationship between the different EDIFACT ontology modules produced by Work Package 7, providing a clear picture of the results provided by the EDIFACT ontologization, as well as the process followed to achieve them. As part of the knowledge conceptualization carried out by the work package, many knowledge source have been used. In encoding meanings from several EDIFACT substandards, the developed set of ontologies have incorporated thousands of terms from external ontologies and their ontological structures.

The EDIFACT ontologies have provided good support for the EAI use case ontology conceptualization. Many business and domain concepts needed by the use case have been reused from the WP7 ontologies. The eHealth use case is based on the encoding of a standardized European Patient Summary. Since there are standards in the field not based on EDIFACT, the developed ontology was based on those standards, not on the EDIFACT ontology. This deliverable describes how the mapping between the EDIFACT and use case ontologies has been done.

Document Information

IST Project Number	FP6 – 027324	**Acronym**	TripCom
Full Title	Triple Space Communication		
Project URL	`http://www.tripcom.org/`		
Document URL			
EU Project Officer	Werner Janusch		

Deliverable	**Number**	7.3	**Title**	Relationships among Ontology Modules within EDIFACT and with External Ontologies
Work Package	**Number**	7	**Title**	Ontological Infrastructure for Business Processes and Data

Date of Delivery	**Contractual**	M30	**Actual**	31-Sept-08
Status	version 0.10		final ☐	
Nature	prototype ☐ report ☒ dissemination ☐			
Dissemination Level	public ☒ consortium ☐			

Authors (Partner)	David de Francisco Marcos (TID), Guillermo López Reyes (TID), and douglas foxvog (NUIG)			
Resp. Author	David de Francisco Marcos (TID)		**E-mail**	davidfr@tid.es
	Partner	TID	**Phone**	+34(983) 367587

Abstract (for dissemination)	This deliverable describes the mappings between the different EDIFACT ontology modules, as well as the mapping with external ontologies (especially OpenCyc) reused during the EDIFACT ontological engineering process. The deliverable documents how the EDIFACT ontology has been used by each of the TripCom use cases, providing examples of EDIFACT knowledge integration in each use case ontology.
Keywords	EDI, EDIFACT, ontology, TripCom, EAI, DAM, eHealth

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
2008-08-07	1	David de Francisco	TOC defined
2008-09-03	2	doug foxvog	Added External Ontology chapter to TOC
2008-09-04	3	doug foxvog	EDIFACT ontology module chapter first draft; External Ontology chapter first draft
2008-09-08	4	doug foxvog	Introduction first draft
2008-08-09	5	Guillermo López	Section 4 filled
2008-09-09	6	doug foxvog	Moved Ontology mapping chapter
2008-12-09	7	David de Francisco	Sections 5 and 6 filled. Abstract and summary added
2008-10-10	8	doug foxvog, David de Francisco	Integration of the comments from the peer review made

Project Consortium Information

Acronym	Partner	Contact
Semantic Technology Institute Innsbruck http://www.sti-innsbruck.at	STI	Prof. Dr. Dieter Fensel Semantic Technology Institute (STI) Innsbruck, Austria E-mail: dieter.fensel@sti-innsbruck.at
National University of Ireland, Galway http://www.deri.ie	NUIG	Dr. Laurentiu Vasiliu Digital Enterprise Research Institute (DERI) Galway, Ireland Email: laurentiu.vasiliu@deri.org
University of Stuttgart http://www.iaas.uni-stuttgart.de/	USTUTT	Prof.Dr. Frank Leymann Inst. für Architektur von Anwendungssystemen (IAAS) Stuttgart, Germany E-mail: frank.leymann@informatik.uni-stuttgart.de
Vienna University of Technology http://www.complang.tuwien.ac.at/	TUW	Prof.Dr. eva Kühn Institut für Computersprachen Vienna, Austria E-mail: eva@complang.tuwien.ac.at
Free University Berlin http://www.ag-nbi.de/	FUB	Prof. Dr.-Ing. Robert Tolksdorf AG Netzbasierte Informationssysteme Berlin, Germany E-mail : tolk@inf.fu-berlin.de
Ontotext Lab, Sirma Group Corp. http://www.ontotext.com/	ONTO	Atanas Kiryakov, Vassil Momtchev, Ontotext Lab, Sirma Group Corp. Sofia, Bulgaria E-mail: vassil.momtchev@ontotext.com
Profium OY http://www.profium.com/	Profium	Dr. Janne Saarela Profium OY Espoo, Finland E-mail: janne.saarela@profium.com
CEFRIEL SCRL. http://www.cefriel.it/	CEFRIEL	Davide Cerri CEFRIEL SCRL. Milano, Italy E-mail: cerri@cefriel.it
Telefonica I+D http://www.tid.es/	TID	Noelia Pérez Crespo Telefonica I+D Madrid, España E-mail: npc@tid.es

Table of Contents

List of Abbreviations

CCR	Continuity of Care Record
CDA	Clinical Document Architecture
DAM	Digital Access Management
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EMEDI	European Medical Electronic Data Interchange
EPS	European Patient Summary
HL7	Health Level 7
LOINC	Logical Observation Identifiers Names and Codes
MILO	MId-Level Ontology
SNOMED	Systematized NOmenclature of MEDicine
SUMO	Suggested Upper Merged Ontology
OWL	Web Ontology Language
UMLS	Unified Medical Language System
UN/EDIFACT	United Nations / Electronic Data Interchange for Administration, Commerce and Transport
URI	Universal Resource Identifier
URL	Universal Resource Locater
XML	eXtensible Markup Language

1 Introduction

The aim of the Ontological Infrastructure work package (WP 7) is to offer a means for a semantically rich definition of business processes for the express purposes of overcoming heterogeneity problems. The concrete goal of Work Package 7 is to create formal ontologies to encode a significant portion of the current EDIFACT Electronic Data Interchange (EDI) standard as a basis for business-to-business process integration.

A standard definition of an *ontology* is a shared, formal conceptualization of a domain[9]. For the purposes of this work package, formal conceptualization means an encoding in a logical computer-processable form and ontologies are restricted to formal definitions of terminology. A *knowledge base* stores information about a particular situation, through encoding that information using one or more ontologies.

Work Package 7 developed ontologies to represent the meanings implicit in EDIFACT messages in Year 2, as described in Deliverable D 7.2[6]. These ontologies represented the range of meanings for the various components of EDIFACT messages, specifically the message types, data segments within those messages, loops of such data segments, data elements within the data segments, and the various codes which were provided as possible fillers for the data elements.

Each set of codes was essentially a separate domain to encode and could be represented by a separate ontology. These ontologies needed to be inter-related, both by defining more general concepts which generalize concepts from different code sets and by covering overlaps of meanings in different code sets. This deliverable is concerned with the coordination of these domains [See Chapter 2].

1.1 Overview of the Rest of the Deliverable

The main objective of this document is to present the extent to which the EDIFACT conceptualization is shared. The sharing exists at multiple levels. Chapter 2 discusses sharing of conceptualizations among the ontologies created for different parts of the EDIFACT standard. Chapter 3 discusses sharing of the conceptualizations derived from EDIFACT and those of ontologies developed external to the TripCom project.

The task of ontologizing portions of the EDIFACT standard was intended to support the work of the use cases, both in determining what is necessary for business messages and in providing a structure for vocabulary for their use. Chapters 4 and 5 discuss this sharing of the developed ontologies with the EAI (WP 8A) and eHealth (WP 8B) use cases respectively Chapter 6 summarizes the results presented in the previous chapters.

2 Relationships among Ontology Modules within EDIFACT

The EDIFACT standard includes scores of message types and data segment types as well as hundreds of data element types. The standard includes sets of codes representing possible values for many of the data element types. As described in D 7.2[6], the ontologization process included encoding the meaning of each of the codes in each of the provided code sets for the data elements in the subset of EDIFACT being ontologized. The range of meanings of each data element, whether designed to be filled with a code set or a text string, was also represented as one or more concepts.

* Start with concept that includes full range of data element
* Use already defined class or relation if found in EDIFACT or external ontology
* Some data elements may have multiple ranges, e.g. one class and one relation
* Create natural subclasses that appear in the file even if not specifically stated
* Define terms (instance/subclass/subrelation) of basic concept for codes in code set
* Define meanings for coded values in terms of the most specific existing term
* Relate term to range for data element
* If an unstated superclass is apt to be used elsewhere, create or link to it
* If groups of element values are naturally clustered, create subclasses/subrelations for each cluster
* If external ontology has intermediate term that provides a possibly useful distinction, include it
* Document term with comment to promote reuse

Table 2.1: Basic Steps for Ontologizing Coded Data Elements

The ontologies for EDIFACT were created modularly to enable users to select appropriate subsets for their purposes, in accordance with the WP7 work plan [2]. In order to keep the ontologies being created consistent, defining relationships among the ontology modules being created was integrated with their construction from the start. Since a decision had been made to use an external ontology to assist in the creation of the EDIFACT ontologies [7], definitions of the relationships between the EDIFACT ontologies and the external ontology were also integral to ontology design. See Table 2.1

In generating the EDIFACT ontologies, the meaning of each term to be handled was determined. This meaning was then checked to determine if it already was defined in one of the already created EDIFACT ontology modules. If not, the external ontology was checked (See Chapter 3.), and if it did not exist there, a term was created in the ontology for the meaning and a definition was specified within the appropriate ontology module in terms of the already existing ontologies.

With multiple people working on developing the ontologies for representing the meaning of EDIFACT messages, a division of the work was necessary. As each code set dealt with a limited range of concepts and most code sets were on quite different topics, each code set was provisionally used as a basis for the creation of a new ontology module. The initial division of labor was to have the same individual work on groups of code sets with similar topics. Some groups of code sets were identified before ontologization work started to be similar enough to merit sharing an ontology; for others, this was not obvious until they were analyzed in more depth. This could be detected in two ways. The first was subjective in that an ontologist who was familiar with one ontology realized that the code set they are starting to work on has similar meanings to the previous one. The analytic method was when the step of searching for the existing meaning or its nearest generalization brought up a term already used in another ontology. If the two terms came from the mainstreams of the different code sets, merging the two ontologies should be considered. If terms from one ontology appeared multiple times when another was being ontologized, then a consolidation to the two ontologies is warranted.

Whenever it turned out that two or more code sets had overlapping or very similar sets of meanings, a single ontology module was used for the similar code sets, with the decision normally being made as work on the second code set was started. If the similarity of the sets of meanings was not detected as the second code set started to be ontologized, the two previously separate modules would be merged into a single ontology when an overlap was detected. Similarly, as the meaning of larger structures such as data segments was analyzed, the relationships between different data elements in the structure needed to be represented in an ontology. The simplest and most natural to do this was often to combine multiple ontology modules.

Combining ontology modules involved both determining concepts which are close common generalizations of terms of the two modules and determining if terms in the two different modules were created for the same concept. In order to ease this process, the more general concepts were normally specified in the original definitions of the terms in their original ontology modules so that inclusion of the same more general concepts would indicate a likelihood of overlap of meanings in the two ontologies.

Upon completion of the first stage on work on the ontologies, the generated ontologies were examined and groups of them were merged as was deemed appropriate.

3 Relationships between EDIFACT Ontology and External Ontologies

Since an ontology is a shared, formal conceptualization of a domain[9], we decided to re-use prior work as much as practical, linking the terms needed to express the EDIFACT concepts to existing or more general terms in public ontologies. Reuse of existing work allowed the relationships among the terms derived from the EDIFACT standards to be more easily identified and helped to prevent multiple terms from being created for the same concepts. To assist in constructing ontologies for representing EDIFACT, a search of preexisting public ontologies was made to find any that already had many of the concepts used by EDIFACT and that had URIs (Universal Resource Identifiers) defined for the concepts.

3.1 Major Broad Public Ontologies

The publicly-available Suggested Upper Merged Ontology (SUMO)[1] has about 7500 terms in a broad ontology with defined OWL URIs. It is extended with a MId-Level Ontology (MILO), and a set of ontologies to total around 20,000 terms. The terms in the extensions do not have defined URIs[13].

Stanford University defined a broad ontology called the TAP Knowledge Base for use on the Semantic Web. It encoded around 65,000 product types, people, and places[10]. URIs, such as `http://tap.stanford.edu/data/Actor` and `http://tap.stanford.edu/AthleteJordan,_Michael`, were defined for each of the terms. The ontology was used by Dill et al.[4] to semantically tag approximately 264 million web pages in 2003. However the ontology is no longer publicly available and its former website[2] is no longer active.

Wikipedia[3] has millions of publicly available web pages describing both individual things and types of things. The URLs of each page can be used as a URI of its referent. However, these terms are not arranged in any hierarchy of relations indicating their membership in a class or subclass relationships between terms. The set of Wikipedia terms is thus not an ontology.

The publicly-available OpenCyc ontology[4] has hundreds of thousands of terms in a broad ontology with defined OWL URIs for each atomic term[5]. It was found to already have many of the concepts used by EDIFACT. A significant portion of its terms have multiple words attached, so that the OpenCyc browser can find a term based on synonymous words and phrases, not just the term name. The Cyc inference engine is complex, permitting reasoning beyond first-order logic. However, the use of terms from the ontology does not require the use of the inference engine or complex features of the language.

On the basis of the above analysis, OpenCyc was selected to as the ontology to use as the upper level of the ontologies created for representing EDIFACT.

[1] http://www.ontologyportal.org/
[2] http://tap.stanford.edu
[3] http://en.wikipedia.org/
[4] http://www.opencyc.org/
[5] OpenCyc also has tens of thousands of functionally defined (and thus non-atomic) terms, which do not have distinct URIs.

3.2 Mapping to External Ontologies

In order to construct the generic ontologies for the EDIFACT messages and tie them together, the terms were defined either directly or indirectly with relation to terms in the external ontology (OpenCyc). The vast majority of the OpenCyc ontology was not used; only those OpenCyc terms used for defining the meanings used in EDIFACT (or in defining the OpenCyc terms used) were included. This resulted in only the appropriate snippets of the external ontology being attached to connect the generated ontologies together.

In generating the EDIFACT ontologies, each term to be generated was checked to determine if it already was defined in the external ontology. This was done using the linguistic information connected to the external term as well as its name.[6] If a more general term was found at this point, its specializations were checked to determine if one of them had the intended meaning.

If the desired meaning existed in the external ontology, the external ontology term was used, with a definition for it being extracted from that ontology. Such definitions were normally simplified, with layers of distinctions present in the external ontology, but not deemed necessary for EDIFACT, ignored.[7] The definitions were made on the basis of common generalizations of the new term and other terms in the developing ontology.

If the meaning from EDIFACT did not exist in the external ontology or one of the already generated ontologies, a new term was defined in terms of the nearest generalization already in the developing or external ontology – whichever was more specific. If this more general term was found in the external ontology, it was considered for inclusion in the developing ontology. If the distinction did not seem necessary for the purposes of encoding EDIFACT, progressively more general concepts were considered, until an appropriate term (or one already being imported into the EDIFACT ontologies) was found. This term was then included in the EDIFACT ontologies.

In cases in which the external ontology had terms which were more specific than that of an EDIFACT term, those external terms which were not already included in the EDIFACT ontology were considered for inclusion in the growing ontology depending on perceived business utility.[8]

In cases in which the EDIFACT concept was not present in the external ontology, but was more general than an external term already used in the ontology, not only was the new concept's definition added to an EDIFACT ontology module, but the definition of the more specific external term was modified to define it relative to the EDIFACT concept.

If a term representing a relationship was being considered for inclusion in the ontology, terms representing the domain and range of relation were also considered. When an external class was adopted into an EDIFACT ontology, relations for which that class was the domain or range were also considered.

External ontology terms which are not generalizations or specializations of those needed for EDIFACT or which represent distinctions deemed immaterial to business messages were not included in the external ontology snippets.

[6] The OpenCyc user interface uses linguistic information attached to each term to provide this.

[7] For example, it was not deemed necessary for the EDIFACT ontology to define Person as a subclass of Hominid, Primate, Eutheria, Mammal, Vertebrate, or Chordate; Animal was sufficient.

[8] AppleComputer might be wanted in the ontology, but AppleG3Computer might not, for instance.

The result was that of over 3400 classes and around 950 relations defined in the EDIFACT ontologies, a little over 1000 of the classes and almost 100 of the relations are OpenCyc terms. In addition, around 2900 "individuals" (items neither classes nor relations; including languages, countries, cities, currencies, and units of measure) from OpenCyc were included in the EDIFACT ontologies, for a total of around 4000 terms.

3.3 Exporting to External Ontologies

As mentioned in Deliverable 7.2[6], the developed ontologies would stand a chance of general reuse if translated into CycL and added to the existing OpenCyc ontology. Such a translation would be a relatively mechanical process.

We suggest that at the end of the ontology refinement step (Deliverable D7.4, due M36) that the definitions of the thousands of new terms defined in this work package be mechanically translated into CycL and submitted for inclusion in the OpenCyc vocabulary.

4 Relationships between the EDIFACT Ontologies and the EAI Use Case Ontology

The goal of the EAI use case [3] is to define a prototype digital content marketplace implementation. The scenario is conceived as a realistic solution for a real business problem, while the functionality of the Triple Space infrastructure is tested. For the semantic description of the use case, an ontology has been developed and refined. Given the business orientation of the use case, the ontology design has reused the work carried out by Work Package 7. The reutilization of EDIFACT-related concepts has enabled easier ontology evolution and maintenance, since most of the knowledge formalized by WP 7 ontologies is widely used by industry.

Although the creation of the EAI ontology is a proof of concept, given the prototypic implementation of the use case, the EDIFACT ontology provides taxonomies that fully define the business domain of the use case, as described below. These EDIFACT ontologies would play a more important role in a commercial implementation of the use case, given that some of the business relationships between Telefónica and its providers use EDI standards.

A small EAI ontology was developed for the initial draft of the use case to match the use case needs. Portions of the EDIFACT ontologies were incorporated as the initial draft was extended where it was necessary to describe knowledge relating commercial aspects already described in the EDIFACT ontologies. However, to complete the EAI ontology, these mappings had to be complemented with newly created use-case-specific terms to specify roles, for example, in the modeled auctioning and cataloging systems or with other general imported concepts relating to the use case, such as concepts relating to types of media products that were not described in EDIFACT.

4.1 Ontology refinement: Mapping to other ontologies

In expanding the first draft of the use case, its ontologies are being mapped to and expanded using the ontologies used and developed in Work Package 7.

The most important ontologies that have been used in WP 8A are the ontologies developed for EDIFACT in D7.2 [6] and the previously existing OpenCyc.

4.1.1 EDIFACT ontologies mappings

The ontologies developed by WP7 define concepts represented by portions of EDIFACT messages or used in them, along with knowledge bases relating these to message components.

The use case ontology has imported commercial concepts from the EDIFACT ontologies, such as bank operations and payment conditions as well as more general concepts such as addresses and other contact information. Portions of taxonomies defined in the EDIFACT ontologies were incorporated *en bloc* as well as individual terms. Many of the terms imported from the EDIFACT ontologies were previously imported from OpenCyc, and there is no distinction in the resulting EAI ontology between those terms and others it directly imported from OpenCyc. In cases where more specific terms than those provided in the EDIFACT ontologies were needed, the ontologies' use of terms from a broad external ontology, eased the use case's finding of such narrower concepts from that ontology. A concrete example in which an EDIFACT taxonomy with OpenCyc concepts and a specific EDIFACT relation are imported can be seen in figure 4.1.

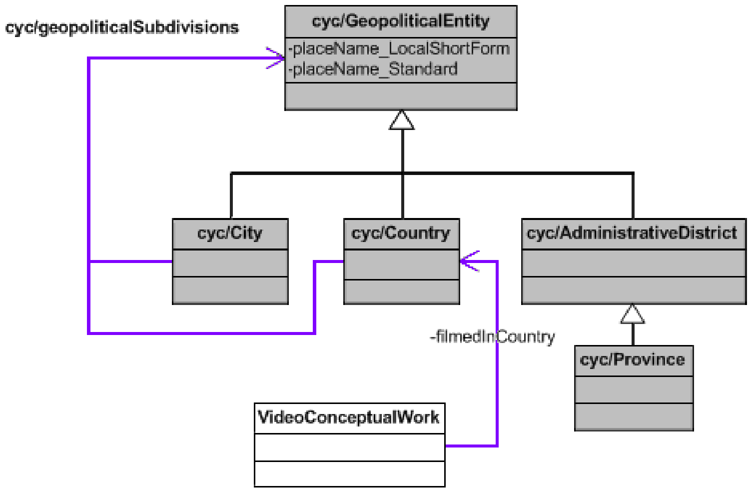

Figure 4.1: An example of the DAM ontology importing a snippet of an EDIFACT ontology taxonomy.

In the use case, a contract defines (among other things) the payment method that will be used in a transaction. The concept `dam/PaymentMethod` is a meta-class, since its instances are types of payment – each of which can have its own instances. The `PaymentMethod` class is defined to serve as the range for the `hasPaymentMethod` slot on `Contract`. Several concepts defined in the EDIFACT BankOperation ontology can be fillers for this slot, including `PaymentByCreditCard`, `PaymentByGiro`, and `PaymentByBankDraft`, as shown in Figure 4.2. The white boxes show DAM-specific concepts, while those imported from the EDIFACT ontologies are shown in grey.

Another example of a mapping is to to the DateFormat ontology. This ontology defines the concept `DateFormat` and defines a set of date formats which are used by various businesses and defined in the EDIFACT standard. These have been included in the DAM ontology.

4.1.2 OpenCyc ontology mappings

Many of the concepts of EDIFACT generic ontologies have been imported from the OpenCyc ontology as described in Chapter 2. The EDIFACT ontologies include and interleave the external concepts with those found only in the EDIFACT standards, often with the more specific terms being newly created for EDIFACT, but the more general term tying them together being extracted from the external ontology. As a result, the restriction on the meaning of a message component is often the external term, while the possible fillers are locally defined.

Many of the EDIFACT ontology terms which WP 8A uses are the more general terms, and thus those referenced from the external ontology.

Since the majority of terms being extracted from the EDIFACT ontologies turned out to be from the external OpenCyc ontology, when desired terms for the WP 8A

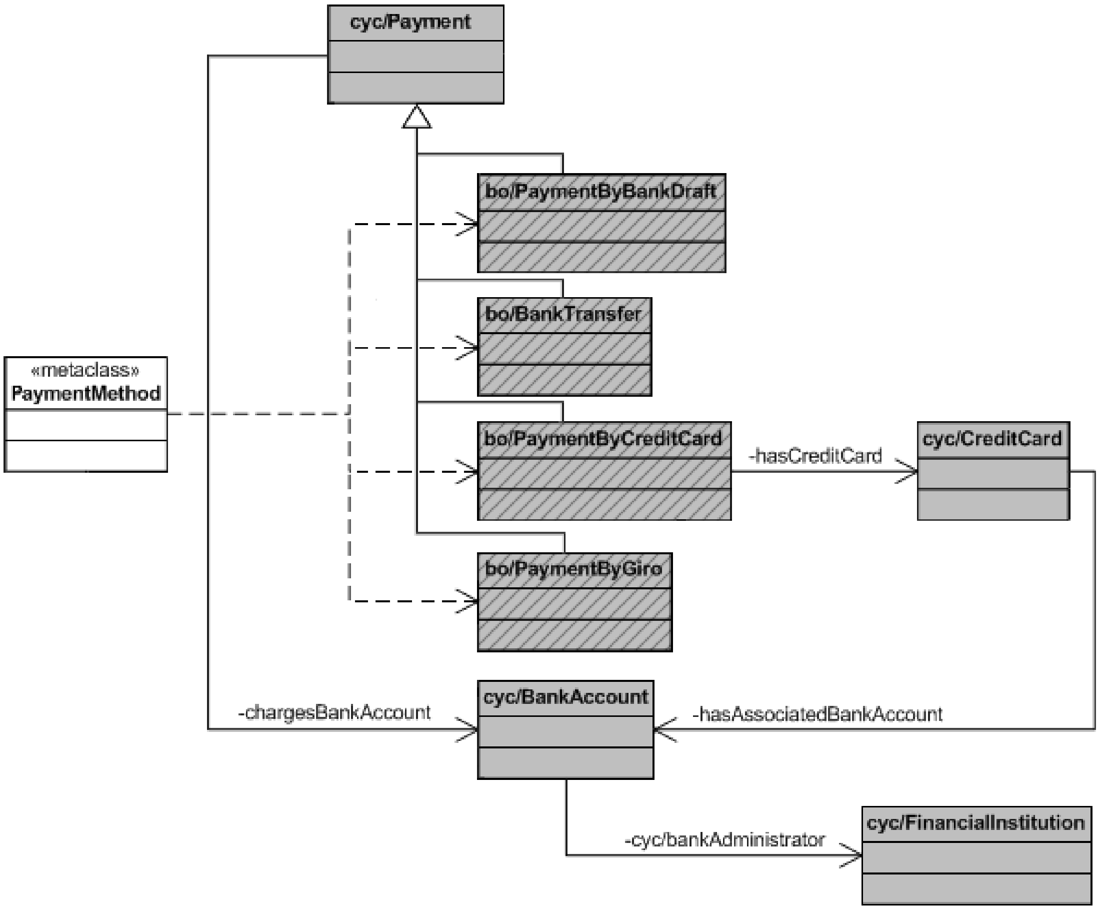

Figure 4.2: An example of the DAM ontology importing a snippet of an EDIFACT ontology.

ontology were not found in the EDIFACT ontologies, OpenCyc was checked. No distinction was made between those OpenCyc terms already in the EDIFACT ontologies and those which were not.

Since the EAI use case final goal is to distribute digital multimedia content, there is a need to describe the content that will be delivered. It is necessary to indicate who the writer of a book is, which songs are sung by a group or singer, and which genre a film belongs to, for example. Building a whole new ontology from scratch to match these knowledge capture requirements would not be efficient, so extensive mappings to OpenCyc (which is one of the largest general knowledge ontologies extant) terms have been made for non use case specific terms. All the mappings to OpenCyc complement and are interleaved with Digital Asset Management (DAM) ontology.

Figure 4.3 shows a small taxonomy of concepts relating musical agents, as composer, player or singer which has been extracted from OpenCyc and integrated into the DAM ontology, by including this taxonomy as a subconcept of `MediaWorkParticipant`: meaning an agent who participates in a `MediaWork` (anything from a song to a book, including films, etc.). The terms are integrated into the DAM ontology with the inclusion of several relations between some of these concepts and others specific to the DAM ontology. Many of these relationships are bi-directional, i.e. there are two relations between the same pair of concepts, one the inverse of the other. Instead of defining both relations, a single relation can be defined, with the inverse relation easily expressed by reversing the argument order. There are also relations among the relationships (super and sub-relationships) which are indicated in the figure with dotted lines. DAM-specific concepts have been marked in white in the figure, while external concepts have been depicted in grey. Similar mappings are made to OpenCyc concepts to include agents such as organizations (businesses, musical groups, etc.) and media professionals (actors, cinema directors, writers, etc.).

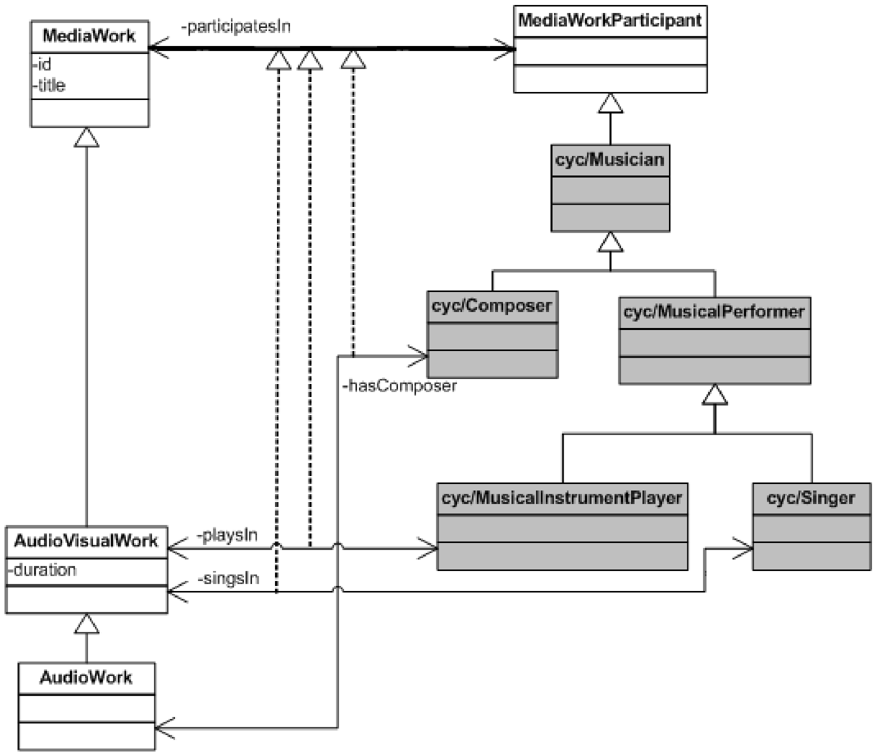

Figure 4.3: An example imported snippet from an external ontology. White concepts are specific to the DAM ontology. Dotted lines show subrelation relationships.

Figure 4.4 shows another example of OpenCyc concepts imported into the DAM ontology. However vast an external ontology may be, there are always areas in which it is necessary to create more precisely defined concepts to fulfill the requirements of the use case. Here, the use case creates several intermediate classes restricted to the types of conceptual works being marketed – distinctions not made in the external ontology. This can lead to generalization graphs in which concepts specific to the DAM ontology are interleaved with selected externally defined concepts, all being integrated into the domain ontology.

DAM ontology areas such as geographic terms and physical addresses have been imported from the EDIFACT ontology selection of OpenCyc terms. An important area in which OpenCyc terms have been imported is financial concepts, such as bank accounting and credit cards. WP 8A expanded on the EDIFACT use of OpenCyc terminology in this area. Other general concepts have also been adopted via the EDIFACT ontologies from OpenCyc, as, e.g., language, date, product, and catalog.

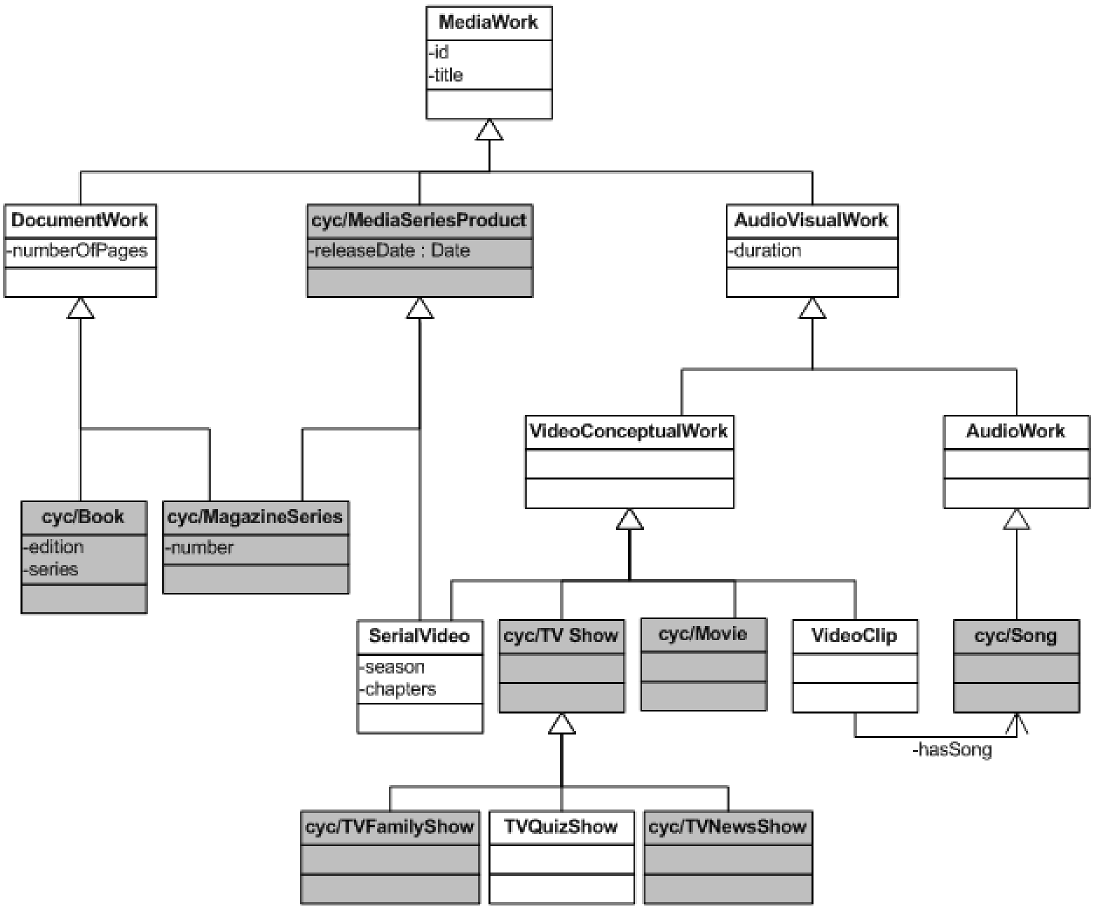

Figure 4.4: An example of integrating external concepts into the DAM ontology.

Subject areas not covered by EDIFACT were also extracted from the OpenCyc ontology. For example, to efficiently describe media work genres, a long list of OpenCyc concepts has been used to instantiate all genres relating music, movies and documents (books and magazines).

Apart from the mapped *classes* of OpenCyc, there are some *relations* that have also been imported into the DAM ontology from OpenCyc, for example `cyc/expiration Date`, which relates a credit card to a date from which it will no longer be valid, `cyc/accountNumber`, which relates a bank account to an account number, `cyc/account Administrator`, which relates a bank account to a bank (`cyc/FinancialInstitution`), `cyc/cityOfAddress`, which relates an address to a specific city, and `cyc/hasMembers`, which relates an organization to a social being which is part of it.

Logically, most of these relationships are not restricted to DAM-specific concepts but to more general concepts. In the context of the use case, however, we can further restrict the argument restrictions of an existing concept – OpenCyc explicitly allows such context-specific restrictions. An example is the relation `cyc/clients`, which relates two social beings, but we restrict to relate a `dam/UseCaseAgent` to a `cyc/Business`.

5 Relationships between EDIFACT Ontology and the eHealth Use Case Ontology

The goal of this section is to describe the relationships between the EDIFACT ontology and that developed for the eHealth use case (Work Package 8B). The eHealth Use Case [12] is focused on the implementation of the European Patient Summary (EPS), an interoperable infrastructure intended to support patient summary management and exchange across European healthcare networks, as an instrument to guarantee an integration of health care across national boundaries. To manage all the knowledge related to this use case, the work package developed ontologies by integrating existing eHealth standards.

These standards define classes of terms used in patient summaries and relations among them, which naturally become the classes and relations defined in an ontology. The standards used are not based on EDIFACT messages, nor are there EDIFACT messages for transmitting patient summaries. However, the concepts that are implicit in the EDIFACT medical messages overlap those used in the patient summary standards, while more general terms in the EDIFACT ontologies are naturally generalizations of the patient summary terms. This allows the EDIFACT ontologies to structure the eHealth ontologies, but does not provide the basic terminology used to define the patient summaries.

The following sections discuss the ontologies used to model the EPS data, the medical coding systems, and the message standards.

5.1 The EPS Ontology

Existing medical applications are based upon eHealth message standards that have been designed to structure and exchange information within a healthcare delivery network. To model the EPS ontology, the structure of extant message standards and medical coding systems were analyzed to model a unified EPS ontology. This ontology formalizes the domain concepts, structures, and relations and associates them with different standardized vocabularies used by various healthcare organizations in different countries. Each of those standards focuses on a specific part of the eHealth domain ranging from exchanging general purpose clinical documents (e.g. the HL7[1]) to modeling patient information relevant for emergency or continuity of care. The CDA (Clinical Document Architecture) and CCR (Continuity of Care Record) standards are based upon XML messages and provide XML schema to describe and validate the structure of the XML message [8]. In addition to the message standards, the medical coding systems formally model the medical terms exchanged in medical messages. These systems define controlled vocabularies and are used in medical records for a variety of purposes, such as coding diseases and medical procedures, recording the causes of death, and for billing and reimbursement.

5.1.1 The CCR and CDA Ontologies

A CCR Message ontology has been built following the Continuity of Care Record (CCR) standard. The CCR standard is a patient health summary standard developed by ASTM International and various US medical societies [11]. It is a way to create flexible documents that contain the most relevant and timely core health information about a patient, and to send these electronically from one care giver to another. The summaries contain various sections such as patient demographics, insurance information, diagnosis and problem list, medications, allergies, and care plan. These represent a "snapshot" of a patient's health data that can be useful or possibly lifesaving, if available at the time of clinical encounter. The CCR standard is designed to permit easy creation by a physician using an electronic health record system at the end of an encounter.

The CDA Message ontology has been built following HL7's Clinical Document Architecture (CDA) standard [5], which is based on the HL7 Reference Information Model. The CDA is designed to be human-readable and is therefore required to contain narrative text, yet still contain structure, and allow for the use of medical codes (such as from SNOMED and LOINC) to represent concepts.

CCR	CDA		EPS		EDIFACT only	EDIFACT terms from OpenCyc
Actor			Actor			SocialBeing
Address			Address			ContactLocation
Age			Age			age(Animal, Time-Quantity)
Alert			Alert			Warning
BrandName			BrandName			BrandNameProduct
CurrentName	Given, Name	hasGiven	CurrentName	hasGiven		nameString(..., _string)
DateTime	BirthTime		DateTime			Date
Description			Description			Description-Statement
Direction			Direction			Instructions
Dose	DoseQuantity		Dose		Dose	-
Duration			Duration			Time-Quantity
Encounter			Encounter			DoctorVisit
FamilyMember			FamilyMember			coreRelatives(Person, Person)
From			From			senderOfInfo(, SocialBeing)
Gender	_string		Gender			PersonTypeByGender
HealthStatus			HealthStatus			PhysiologicalCondition
ExtendedID			IDs			identificationStrings(..., _string)
Immunization			Immunization			VaccinationEvent
Indication			Indication		Symptom	symptomOfAilment(... PhysiologicalCondition)
InformationSystem			InformationSystem			ComputationalSystem
Instruction			Instruction			Instructions
Interval			Interval			Time-Quantity
Language			Language			HumanLanguage
Location			Location			Place
Medication			Medication			DrugProduct
Name			Name			nameString(..., _string)
Organization			Organization			Organization
Payer			Payer			payer(..., SocialBeing)
Practitioner			Practitioner			MedicalCareProfessional
Problem			Problem			PhysiologicalCondition
Procedure			Procedure			Medical Procedure
Product			Product			Product
ProductName	Consumable		ProductName			productTypeName(..., _string)
Provider			Provider			MedicalCareProvider
Quantity			Quantity			Quantity
Result			Result		ResultValue	
Severity			Severity			Severity
Status			Status			ObjectEventStatus
Strength			Strength		productStrength	
SupportProvider			SupportProvider			agentSupportsAgent-Generic
Telephone			Telephone			agentPhoneNumber
Test			Test			MedicalTesting
TestResult			TestResult		ResultValue	
			CodingSystem		CodeSet	
	Center		Measure			Quantity
	PatientPatient		Person			MedicalPatient, Person

EDIFACT uses relation	In OpenCyc; not EDIFACT
Possibly erroneous mapping	OpenCyc uses relation; not in EDIFACT

Figure 5.1: Extract of the EPS-EDIFACT Mappings

5.2 Mapping to the EDIFACT Ontologies

Many of the terms used in the CCR and CDA ontologies are equivalent to those used in the EDIFACT ontologies, either directly created for EDIFACT or incorporated from OpenCyc. Although they were not created on the basis of the EDIFACT ontologies, mappings have been established between the EPS and the EDIFACT terms.

Some classes defined in the WP 8B ontologies (such as `FamilyMember`, `Payer`, and `ProductName`) are modeled in EDIFACT as relations (correspondingly: `coreRelatives`, `payer`, and `productTypeName`) as their meanings had been judged in the EDIFACT ontologization process to not be a type of thing, but a relationship between two things. WP 8B is considering altering its ontology in such cases.

Figure 5.1 lists the mappings determined between the terms used in the EPS Use Case and those used in the ontology created for the EDIFACT standards or the external ontology it used. The cases in which the EDIFACT ontology uses a relation are indicated in green. Most of the terms from the external ontology used in this mapping were ones already incorporated in the EDIFACT ontologies; only four EPS classes were equivalent to OpenCyc classes not included in EDIFACT (indicated in yellow) and one non-EDIFACT OpenCyc relation (indicated in purple) was mapped to. The EDIFACT ontologies do not include the concept "`cyc/PersonTypeByGender`" because it is a meta-class, but do include its instances, `cyc/MaleHuman` and `cyc/FemaleHuman`.

WP 8B does not need to expand its ontology at the moment, nor does it discern a need to change its class names from the names of the slots in the CCR and CDA message standards. However, the generated mapping of the EPS terms into those of the EDIFACT ontologies enable easier expansion and provide a basis for future reasoning over patient records.

6 Conclusions

The EDIFACT ontologies developed within the TripCom WP7 capture the EDIFACT message formats, structure and semantics. The amount of knowledge taken into account in this work has brought about a great amount of ontology modules, whose structure is documented in D7.2 [6] and this deliverable. External general-purpose ontologies, mainly OpenCyc, have been used to align and complete the EDIFACT knowledge. The mapping with these ontologies have been included in the deliverable in order to provide a complete and uniform view of the knowledge structure built.

As part of the objective of documenting the EDIFACT relationships with external ontologies, an overview about the usage of the EDIFACT ontologies being made by each use cases have been also provided. While the EAI use case is focused on a clear business domain, the eHealth use case has needed to focus on several widely used medical standards. This has resulted in a strong integration of EDIFACT-related concepts by the EAI use case ontology, which have been described. The eHealth ontology, for reasons discussed above, had to be based on and aligned with the different medical standards. Those standards are discussed above along with the alignment of the derived ontology with that derived from EDIFACT.

We recommend that the ontologies created from the EDIFACT system be semi-mechanically converted into CycL and submitted for inclusion in OpenCyc in order to encourage the reuse of the developed ontologies. OpenCyc is managed by The Cyc Foundation with a charter to "grow the ontology and knowledge base ... with the help of volunteers". The contribution of a developed CycL ontology should be straightforward.

References

[1] G. Beeler and et al. HL7 Version 3 Message Development Framework, 2006.

[2] TripCom Consortium. Annex i; description of work. Technical report, TripCom, 2005.

[3] David de Francisco Marcos, Lyndon Nixon, and Germán Toro del Valle. Towards a multimedia content marketplace implementation based on triplespaces. In A. Sheth et al., editor, *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, number 5318 in Lecture Notes in Computer Science, pages 875–888. Springer-Verlag Berlin Heidelberg, October 2008.

[4] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 178–186, New York, NY, USA, May 2003. ACM.

[5] Robert H. Dolin, Liora Alschuler, Calvin Beebe, Paul V. Biron, Sandra Lee Boyer, Daniel Essin, Elliot Kimber, Tom Lincoln, and John E. Mattison. The HL7 Clinical Document Architecture. *J Am Med Inform Assoc*, 8(6):552–569, 2001.

[6] doug foxvog. Ontology of edifact syntax and semantics; tripcom deliverable d7.2. Deliverable, NUIG, October 2008.

[7] doug foxvog, Juan Pablo Palacios, and Elena Paslaru Bontas Simperl. Analysis of edifact and other standards; tripcom deliverable d7.1. Deliverable, NUIG, September 2006.

[8] Jeffrey Ferranti, Clayton Musser, Kensaku Kawamoto, and Ed Hammond. The Clinical Document Architecture and the Continuity of Care Record: A Critical Analysis. *J Am Med Inform Assoc*, pages 245–252, May-Jun 2006.

[9] Tom Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5:199220, 1993.

[10] Ramanathan Guha and Rob McCool. TAP: a Semantic Web platform. *Computer Networks*, 42:557–577, August 2003.

[11] D.C. Kibbe, Jr. R.L. Phillips, and L.A. Green. The continuity of care record. *Am Fam Physician*, 70(7):1222–3, 2001.

[12] Reto Krummenacher, Elena Paslaru Bontas Simperl, Lyndon J. B. Nixon, Dario Cerizza, and Emanuele Della Valle. Enabling the european patient summary through triplespaces. In *CBMS*, pages 319–324. IEEE Computer Society, 2007.

[13] Adam Pease. Private communication, 2008.