Some Issues on Ontology Integration*

H. Sofia Pinto
Instituto Superior Técnico
Departamento de Eng. Informática
Grupo de Inteligência Artificial
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Tel: (351-1) 8417641 Fax: (351-1) 8417472
sofia@gia.ist.utl.pt

Asunción Gómez-Pérez
Laboratorio de Inteligencia Artificial
Facultad de Informática
Campus de Montegancedo sn.
Boadilla del Monte, 28660, Madrid, Spain
Tel (34-1) 3367439 Fax: (34-1) 3367412
asun@delicias.dia.fi.upm.es

João P. Martins
Instituto Superior Técnico
Departamento de Eng. Informática
Grupo de Inteligência Artificial
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Tel: (351-1) 8417472, Fax: (351-1) 8417472
jpm@gia.ist.utl.pt

Abstract

The word *integration* has been used with different meanings in the ontology field. This article aims at clarifying the meaning of the word "integration" and presenting some of the relevant work done in integration. We identify three meanings of ontology "integration": when building a new ontology reusing (by assembling, extending, specializing or adapting) other ontologies already available; when building an ontology by merging several ontologies into a single one that unifies all of them; when building an application using one or more ontologies. We discuss the different meanings of "integration", identify the main characteristics of the three different processes and propose three words to distinguish among those meanings: *integration*, *merge* and *use*.

1 Introduction

Within Knowledge Sharing and Reuse, the field of Ontological Engineering (OE) is an active area of research. One of its open research topics is ontology integration. Unfortunately, there has been an abusive use of the word *integration* within the community. Integration designates, not only, the special operations to build ontologies from other ontologies available in some ontology development environments (Farquhar, Fikes & Rice 1997), but also the process of building ontologies from other preexistent ontologies (Borst, Akkermans & Top 1997, Dalianis & Persson 1997, Gangemi, Pisanelli & Steve 1998, Skuce 1997, Swartout, Patil, Knight & Russ 1997), the set of activities within some methodologies that specify how to build ontologies using other publicly available ontologies (Uschold & King 1995, Gruninger 1996, Fernández, Gómez-Pérez & Juristo 1997), the use of ontologies in applications (Bernaras, Laresgoiti & Corera 1996, Uschold, Healy, Williamson, Clark & Woods 1998), just to name a few. Integration in ONIONS (Gangemi et al. 1998), doesn't mean the same as in the Ontolingua Server (Farquhar, Fikes, Pratt & Rice 1995) or in PhysSys (Borst 1997).

This article aims at clarifying and characterizing the several meanings of the word *integration*. This article is organized as follows. Section 2 identifies the three meanings usually associated to the word and proposes three words to

*This work was partially supported by JNICT grant No. PRAXIS XXI/BD/11202/97 (Sub-Programa Ciência e Tecnologia do Segundo Quadro Comunitário de Apoio) and project PRAXIS XXI/1568/95.

Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999

(V.R. Benjamins, B. Chandrasekaran, A. Gomez-Perez, N. Guarino, M. Uschold, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/

refer to those meanings in the OE field. Sections 3, 4 and 5 discuss the differences between those meanings. In each one of those sections we review some of the most relevant work, we characterize each one of the different processes and present our views and conclusions. Section 6 emphasizes our main conclusions.

2 What Have We Been Calling Integration?

We identified three different situations in which the word integration has been used:

1. *Integration of ontologies when building a new ontology reusing other available ontologies.* In this case one wants to build a new ontology, there are some ontologies built and made available which are parts of the new ontology and such ontologies match the appropriate requirements. For instance, we want to build an ontology on Control Systems and EngMath (Gruber & Olsen 1994) satisfies our requirements for it, as for instance, adequate levels of detail and granularity, it is implemented in an adequate language[1], etc. Since this ontology is publicly available it should be reused. In some cases a whole ontology can be built just from assembling other ontologies. Some other times the reused ontologies must be extended, specialized or adapted. So, ontologies are reused to build new ones.

2. *Integration of ontologies by merging different ontologies about the same subject*[2] *into a single one that "unifies" all of them.* In this case, one wants to build an ontology merging ideas, concepts, distinctions, axioms, etc., that is knowledge, from other existing ontologies on exactly the same subject. For instance, among the considerable number of medical ontologies there is the Unified Medical Language System (UMLS) (Humphreys & Lindberg 1992) and the Galen COding REference (CORE) model (Rector, Gangemi, Galeazzi, Glowinski & Rossi-Mori 1994). There are differences between them, not only in the basic distinctions[3] but also in the way those terms are defined (in the meaning behind those terms), in the representation ontologies (van Heist et al. 1997) used to implement them, etc. When all these different ontologies are "integrated", in the sense of merged, unified, a new ontology about the medical domain is built. This ontology tries to unify concepts, terminology, definitions, constraints, etc., from all of them and, if implemented, using a particular representation ontology. So, ontologies are merged, unified into a single one.

3. *Integration of ontologies into applications.* In this case, one wants to introduce into an application one or more ontologies that underly and are shared among several software applications or one uses one or more ontologies to specify or implement a knowledge based system (KBS). For instance, if we want to build an application on airplanes there is a lot of knowledge about airplanes in general (what parts are they built from, how are they designated, how do they interact, what laws determine the way those parts work, etc.) that needs to be formalized and implemented. The knowledge that is not specific to any particular airplane should be represented in an ontology. When we build another application that also needs knowledge about airplanes we should use the ontology already built (or adapt it if necessary). This airplane ontology will, once again, underlie an application. So the ontology(ies) is (are) used or reused to build an application.

From now on, we will refer to each kind of integration as *integration*, *merge* and *use*, respectively.

3 Integration

Although some methodologies to build ontologies acknowledge the need for an integration step or the importance of integration activities in the process of building an ontology, the important problems of integration (how should integration be performed?, etc.) remain more or less unsolved.

3.1 Tools That Allow Integration

The work on the Ontolingua Server (Farquhar et al. 1995, Farquhar, Fikes & Rice 1996, Farquhar et al. 1997), an ontology development environment for collaborative ontology construction, addressed the problem of ontology integration. This tool allows collaborative ontology building and also provides an ontology library, where tested ontologies are gathered and made publicly available. To allow reuse of the ontologies made available at the Ontolingua Server library, a set of integration operations was identified, specified, defined and made available to ontology builders. Users are allowed three operations (Farquhar et al. 1997): inclusion, polymorphic refinement and restriction (specialization). *Inclusion* is used when the ontology is included (from the library of ontologies kept by the tool) and used as it is. The inclusion relations between ontologies may be circular, so one concept in one ontology can point to a concept in another ontology that, again, points to another

[1] Either the required language, or there are good translators available between the language in which it is implemented and the required language.

[2] By subject we mean what the ontology deals about. We avoid the term domain since it is only used to describe domain specific ontologies and this kind of integration can be used to build general (generic (van Heist, Schreiber & Wielinga 1997), top-level (Guarino 1998) or upper-level) ontologies.

[3] Classification criteria imposed on the terms gathered and defined/described in those ontologies, that is concept classifications.

concept in the first ontology. *Polymorphic refinement* extends one operation so that it can be used with several kinds of arguments. *Restriction* makes simplifying assumptions that restrict the included axioms. The Ontolingua Server also provides facilities for local symbol renaming. This facility enables ontology developers (1) to refer to symbols from other ontologies using names that are more appropriate to a given ontology and (2) to specify how naming conflicts among symbols from multiple ontologies are to be resolved. All these simple integration operations have been proposed to allow some sort of ontology integration in the Ontolingua Server.

3.2 Ontologies Built Through Integration

The Physical Systems ontology (PhySys) (Borst, Benjamin, Wielinga & Akkermans 1996, Borst et al. 1997, Borst 1997) is based on the reuse of the EngMath ontology (Gruber & Olsen 1994), and on general ontologies such as Mereology, Topology, Systems Theory, Component and Process (Borst et al. 1996), which were implemented in Ontolingua (Gruber 1993). To allow reuse, some general integration operations[4] were identified, specified and defined. For instance, the Mereology ontology is reused (more precisely, extended) by the Topology ontology, The operations allowed are (Borst 1997): include and extend, include and specialize and include and map[5]. With *include and extend* the "imported ontology is extended with new concepts and relations". With *include and specialize* an "abstract theory is imported and applied to the contents of the importing ontology". With *include and map* "different viewpoints on a domain are joined by including the views in the domain ontology and formalization of their interdependencies". Since this operation contains a lot of domain knowledge it is considered an ontology on its own. In (V., Gomez-Perez, T. & Pinto 1998) we present the Reference Ontology that was incorporated into the $(KA)^2$ ontology (Benjamins & Fensel 1998), more precisely into the Product subontology.

There can be some less formal and clean ways of ontology integration. For instance, (Dalianis & Persson 1997) describes the construction of an ontology for electrical distribution networks. This ontology was built by reusing an ontology for electrical transmission networks (Bernaras et al. 1996), more precisely its structural subontology. The ontology developed for electrical distribution networks is also a structural ontology. Although the domains are not the same and some adaptations had to be performed, the percentage of reused concepts was quite high. This hints that perhaps a more general ontology could be defined. The KACTUS (Schreiber, Weilinga & Jansweijer 1995) toolkit was used to edit the pre-existent ontology (available from the KACTUS ontology library) and create the electrical distribution ontology.

Sometimes it may be possible to mistakenly take integration for maintenance activities if we just want to improve or slightly modify the integrated ontology. For instance, in the case of CHEMICALS (Fernández 1996, Fernández et al. 1997) the length centimeter ("cm") unit, a length unit commonly used in Europe (but not in the USA), was needed. The Standard-Units ontology (Gruber & Olsen 1994) available in the Ontolingua Server library did not include such unit when CHEMICALS was implemented in the Ontolingua Server. The solution found, with the operations available, was: to develop a new ontology which included Standard-Units and add to it the needed unit. However the right solution, adopted latter, was the inclusion of this unit in the Standard-Units ontology kept in the library. This is the appropriate solution since this is not a specific purpose unit, but a world wide generally accepted one and it applies to all domains that may reuse the Standard-Units ontology.

3.3 Methodologies That Include Integration

The methodology to build ontologies presented in (Uschold & King 1995) includes an integration step. This methodology proposes that integration should be done either during capture (knowledge acquisition), coding (implementation) or both. However no solutions for the problem of how integration is done are proposed or discussed. The problem is only recognized as a difficult one.

The methodology to build ontologies proposed in (Gruninger 1996) also refers integration. This methodology mentions two kinds of integration: "combining ontologies that have been designed for the same domain" and "combining ontologies from different domains". Once again the problem of integration is considered difficult since two ontologies may use the same terminology with different semantics. According to this methodology, ontologies are built based on ontology building blocks and foundational theories. According to the building blocks and foundational theories of the ontologies being integrated, integration is distinguished into: integration (at the level) of the building blocks, the most simple; integration (at the level) of the foundational theories, which is more difficult and may result in only partial integration; and ontology translation when the ontologies are so different that they share neither the building blocks nor the foundational theories, which makes integration extremely difficult.

METHONTOLOGY (Fernández et al. 1997, Blázquez, Fernández, García-Pinar & Gómez-Pérez 1998, Fernández, Gómez-Pérez, Sierra & Sierra 1999) is another methodology to build ontologies that also considers integration. It proposes that the development of an ontology should follow an evolving prototyping life cycle and not a waterfall one. Although in earlier versions integration was consid-

[4] They were named projections. These projections formalize the dependencies between concepts and relations in different ontologies.

[5] In earlier versions it was named include and project.

ered as a state during the development of an ontology (after formalization and before implementation), recent versions consider it as an activity (as well as knowledge acquisition and evaluation (Gómez-Pérez, Juristo & Pazos 1995)) that should be performed since specification until maintenance. This methodology proposes that ontology building, and therefore ontology integration, should be done preferably at the knowledge level (Newell 1982) (in conceptualization) and not at the symbol level (in formalization, when selecting the representation ontology) or at the implementational level (when the ontology is codified in a target language).

3.4 Our View

In integration we have, on one hand, one (or more) ontologies that are *integrated* ($O_1, O_2, ..., O_n$, Figure 1), and on the other hand, the ontology *resulting* from the integration process (O, Figure 1). The integrated ontology(ies) are those that are being reused. They are a part of the resulting ontology. The ontology resulting from the integration process is what we want to build and although it is referenced as one ontology it can be composed of several "modules", that are (sub)ontologies. This happens not only in integration but when building an ontology from scratch. For instance, the Enterprise ontology (Uschold, King, Moralee & Zorgios 1998) is composed by several modules, called sections, like Meta-Ontology (where concepts like entity, relationship, role, etc. are represented), Activities and Processes (where concepts like activity, resource, plan, etc. are represented), Organization, etc. However we call the resulting ontology the Enterprise ontology and usually do not refer to its parts unless we specifically want to talk about them.

The domain of the integrated ontology is different from the domain of the resulting ontology but there may be a relation between both domains. When the integrated ontology is reused by the resulting ontology, the integrated concepts can be, among other things, (1) used as they are, (2) adapted (or modified), (3) specialized (leading to a more specific ontology on the same domain) or (4) augmented by new concepts (either by more general concepts or by concepts at the same level). The domains of the different integrated ontologies usually are different among themselves, that is, each ontology integrated in the resulting ontology usually is about a different domain either from the resulting ontology (D, Figure 1) or the various ontologies integrated ($D_1, D_2, ..., D_k$, where usually k = n, Figure 1). In integration, the resulting ontology should be such that there is no similar ontology already built, otherwise one should simply reuse the existing one.

In integration one can identify regions in the resulting ontology that were taken from the integrated ontologies. Knowledge in those regions was left more or less unchanged. In the example presented in Figure 2, one

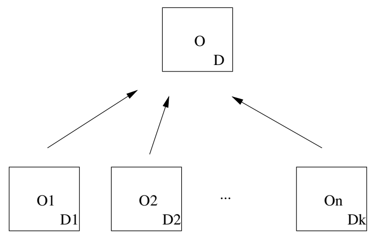

Figure 1: Integration

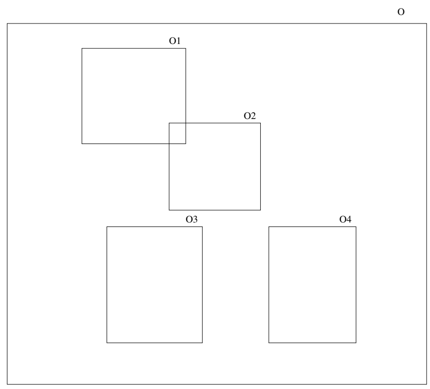

Figure 2: The resulting ontology in an integration process

can identify in the structure of the resulting ontology (O, Figure 2) the four ontologies that were integrated (O_1, O_2, O_3, O_4, Figure 2) to build it. In this case, more knowledge had to be added to form the resulting ontology, besides knowledge coming from the integrated ontologies.

The ontologies reused are chosen from those available in ontology libraries that meet a series of requirements, for example, domain, abstraction, type (van Heist et al. 1997), generality, modularity, evaluation, just to name a few. The resulting ontology should have all the properties of a good ontology. Not only should it be clear, coherent, extensible, comply to the principle of the minimal ontological commitment[6] and to the minimal encoding bias[7], as proposed in (Gruber 1995) but it should also be complete, concise, non-ambiguous (related to coherence), have an adequate level of detail, be built upon the appropriate basic distinctions, have been evaluated, etc.

The problem is how to integrate several existing ontologies within a new one that is being built. Problems such as consistency of the resulting ontology, level of detail throughout the whole ontology[8], etc, have to be dealt

[6] Only describe the vocabulary needed to talk about the domain.

[7] In conceptualization one should only consider the knowledge level; therefore choices for purely implementational convenience should be avoided.

[8] That is, the ontology doesn't have "islands" of exaggerated level of detail and other parts with an adequate one. It should be stressed that none of the parts should have less level of detail than the one required or else the

with. The solution seems to be the specification of a set of integration operations that tell how knowledge in the integrated ontology is going to be included and combined with the knowledge in the resulting ontology. Integration operations can be viewed as composing, combining or assembling operations. However these operations should only be performed if the integrated ontologies have a series of features. Not only will the features assure that the integrated ontology is the most appropriate one but also that the integration operations can be successfully applied and that the resulting ontology will have the desired characteristics. In (V. et al. 1998) we present a series of features (and a WWW broker), not specifically for integration purposes, that can help the search for suitable ontologies. Comparing both sets of the proposed integration operations we can see that "restriction" (Farquhar et al. 1997) and "include&specialize" (Borst 1997) work similarly. As described in (Borst 1997), both "include&extend" and "include&map" are abstract operations and can be performed in the Ontolingua server with the available operations. However a larger set of integration operations needs to be identified, specified and defined.

As the development of an ontology should follow an evolving prototyping life cycle, the ontology may be considered for integration in specification, conceptualization, formalization, implementation and maintenance. That is, we can have different integration procedures for the same ontology but in different states of the ontology building process. However the effort of integration varies: it is more significant in the earliest states (specification and conceptualization) than in the final ones (after implementation). As we have an evolving prototyping ontology building life cycle the same ontology can be used in the same state in integration activities more than once. These procedures and activities form the overall process of integration. The integration process needs to be further studied, namely, the integration procedures and activities need to be defined.

4 Merge

Merge is the issue where more work has been developed, so far. There is a wide variety of projects from a wide range of domains, for example development of natural language ontologies, like SENSUS (Knight & Luk 1994), or ontologies on the medical domain like UMLS (Humphreys & Lindberg 1992); the search for agreed upper level ontologies, like (Skuce 1997), (Guarino 1997) or (Sowa 1995); the search for merge methodologies in the medical domain, ONIONS (Gangemi et al. 1998).

4.1 Ontologies Built Through Merge

SENSUS (Knight & Luk 1994, Knight, Chander, Haines, Hatzivassiloglou, Hovy, Iida, Luk, Withney & Yamada 1995, Swartout et al. 1997) (a natural language ontology) was built by extracting and merging information from existing electronic sources. Several existing resources were used since each one had different and important features: PENMAN Upper Model (Bateman, Kasper, Moore & Whitney 1989), ONTOS, WordNet (Miller 1990) and an electronic natural language dictionary. The PENMAN Upper Model and ONTOS are high-level linguistically-based ontologies but lack a broad coverage of terms. They provided the upper level-organization. WordNet is a thesaurus-like hierarchically organized semantic network lacking the upper level structure with broad coverage of terms. It provided the middle structure and terms. Finally, the electronic natural language dictionary, with both broad coverage of words and Semantic Categories,[9] provided both terms and the upper level organization.

There have also been several efforts to find a top-level ontology that could find the agreement of a broad number of researchers and of systems. Among these "merged" upper-level ontologies, we briefly describe Skuce's (Skuce 1997) and Guarino's (Guarino 1997) approaches since they follow very different processes to try to achieve the same purpose: an upper-level ontology. Both upper-level ontologies are unfinished proposals. Skuce (Skuce 1997) tries to find, at least, one of what is called an Agreed-Upon-Ontology. This ontology, a rather general one, defines the most general Fundamental Ontological Distinctions - FOD's which any top-level (general and domain-independent) ontology should have. In (Skuce 1997), he presents the definitions of concepts like ontology, primitive, distinction, category and entity and then presents the fundamental distinctions that he finds important. Some of them are concrete/abstract; atomic/composite; material/place; discrete/continuous; state/process; dependent/independent; and instance/predicate.

Guarino's approach (Guarino 1997) is based on the solid grounds provided by philosophers that have been addressing these issues for the past 2000 years. His study on the ontological distinctions issues is organized around a theory of parts, a theory of wholes, a theory of identity, a theory of dependence and a theory of universals. From these theories he defines a preliminary taxonomy of top-level ontological concepts that combines clarity with semantic rigor, generality and common sense. His taxonomy of top-level ontological concepts is divided into an ontology of Particulars and an ontology of Universals. The backbone of distinctions of particulars around sortal categories considers substract, object and quality. The non-sortal categories include Mereological, Physical, Functional, Biological, Intentional and Social Stratums. Concrete objects have yet two other sets of distinctions: singular and plural; and body and feature. (Unary) Universals have two basic distinctions: tax-

ontology would be useless, since it would not have sufficient knowledge represented.

[9] Which associates certain word senses with particular fields (medicine, biology, etc.).

ons and properties. A more detailed description can be found in (Guarino 1997).

4.2 Methodologies To Perform Merge

Let us first present how SENSUS was built. First PENMAN, ONTOS and the Semantic Categories of the electronic natural language dictionary were merged, by hand. This produced the ontology base (the upper level of the ontology). Then, WordNet was merged into this base, again by hand. Finally, a semi-automatic tool helped merging WordNet with the English Dictionary by matching similarities in the textual definitions and by using the hierarchical organization of WordNet. This final merge was finally included in the ontology. In each merge step more than one ontology or source of knowledge was considered at the same time. However, the merge process was subdivided for each level of the ontology (beginning in the upper ones and ending on the lower ones). In an effort to ease and search for a proper methodology to do this kind of merge process, Hovy (Hovy 1996) tried to identify a set of relevant features that should be considered when comparing different ontologies for the purpose of merging ontologies, in particular, natural language ones.

The methodology followed by Skuce to find the ontological distinctions presented in (Skuce 1997) was brainstorming, followed by meetings with other researchers interested in the problem. The work is still incomplete, so he suggests to incrementally build from the list presented. The proposed methodology begins with the creation of a group involving a diverse group of researchers working in different locations. Each member should develop a list of primitives, distinctions and categories (a classification of FOD's according to the way they are defined) that should be carefully chosen, defined and carefully documented (choices and definitions). The choices are presented to the group for discussion and approval. Only when they are agreed upon can they get to the formalization stage. There can be several iterations of the previous steps. The agreed proposals are presented to wider audiences for criticism. Some can go back to the initial stages, some may be ready to be accepted. The idea is to try to find a standardized upper-model that would greatly ease some kinds of integration efforts.

ONIONS (ONtologic Integration Of Naive Sources) (Gangemi, Steve & Giacomelli 1996, Steve & Gangemi 1996, Gangemi et al. 1998) is a methodology for merging[10] ontologically-heterogeneous taxonomic knowledge which has been used to build the formal medical ontology IMO (Integrated Medical Ontology) and a library of generic ontologies ON9. The ONIONS methodology was successfully applied to five medical sources: UMLS (Humphreys & Lindberg 1992, Humphreys & Lindberg 1993), a medical ontology implemented in a semantic network; SNOMED-III (Cote, Rothwell & Brochu 1994) and GMN (Gabrieli 1989) partial hierarchical ontologies; ICD10 classification (WHO 1994), which is a hierarchical ontology; and CORE model (Rector et al. 1994) developed under the GALEN project (Rector, Solomon & Nowlan 1995)[11]. In the medical domain most available ontology sources are just taxonomies (UMLS is the exception). The ONIONS 6-step methodology can be summarized as: (1) Analyzing and selecting the relevant sets of terms from the various terminological sources. All sources are considered in this step. (2) Finding local definitions of the terms by analyzing the classification criteria used to define the terms. (3) Finding the theories related to the distinctions made in the local definitions, that is one tries to find the general (global) ontologies (in contrast to the surface ontologies associated to the local definitions found in the previous step). For that, one has to try to find theory chunks if no theories are available. (4) Finding the theories for the top-level design using the same procedures as in the previous step. The top-level categories found depend on the "taste" of the ontological engineer, so the proposed taxonomy should be taken as a possible choice and easily modifiable. (5) Merging local definitions with the top-level categories. One tries to find the direct correspondences among local items and elements of the theory chunks found for the top-level or amends/enlarges theory chunks to allow local items to have room according to the proposed top-level. (6) The model is formalized, and eventually, implemented. In (Steve & Gangemi 1996), the ontological commitments of this methodology are favorably analyzed (a-posteriori) through the ontological commitment rules proposed in (Guarino, Carrara & Giaretta 1994). It would be interesting to see how general the methodology is, by trying to apply it to other domains. Its authors claim that their methodological choices restrict the currently feasible development of formal ontologies to the merge of explicit task-oriented expert knowledge.

4.3 A Definition of Merge

Sowa (Sowa 1997) defined merge as:

> The process of finding commonalities between two different ontologies A and B and deriving a new ontology C that facilitates interoperability between computer systems that are based on the A and B ontologies. The new ontology C may replace A or B, or it may be used as an intermediary between a system based on A and a system based on B. Depending on the amount of change necessary to derive C from A and B, different levels of "integration" can be distinguished: alignment, partial compatibility, and unification. Alignment is the weakest form

[10] The authors use the word integration.

[11] The researchers of ONIONS were also involved in the GALEN project.

> of "integration": it requires minimal change, but it can only support limited kinds of interoperability. It is useful for classification and information retrieval, but it does not support deep inferences. Partial compatibility requires more changes in order to support more extensive interoperability, even though there may be some concepts or relations in one system or the other that could create obstacles to full interoperability. Unification or total compatibility may require extensive changes or major reorganizations of A and B, but it can result in the most complete interoperability: everything that can be done with one can be done in an exactly equivalent way with the other.

In the last quotation the word "integration" should be understood as merge. He further defines alignment as:

> Alignment: A mapping of concepts and relations between two ontologies A and B that preserves the partial ordering by subtypes in both A and B. If an alignment maps a concept or relation x in ontology A to a concept or relation y in ontology B, then x and y are said to be −equivalent−. The mapping may be partial: there could be many concepts in A or B that have no equivalents in the other ontology. Before two ontologies A and B can be aligned, it may be necessary to introduce new subtypes or supertypes of concepts or relations in either A or B in order to provide suitable targets for alignment. No other changes to the axioms, definitions, proofs, or computations in either A or B are made during the process of alignment. Alignment does not depend on the choices of names in either ontology. ...

He further defines partial compatibility as:

> Partial compatibility: An alignment of two ontologies A and B that supports equivalent inferences and computation on all equivalent concepts and relations. If A and B are partially compatible, then any inference or computation that can be expressed in one ontology using only the aligned concepts and relations can be translated to an equivalent inference or computation in the other ontology.

He further defines unification as:

> Unification: A partial compatibility of two ontologies A and B that has been extended to a total compatibility that includes all concepts and relations in both A and B. If the ontologies of A and B have been unified, then any inference or computation that can be expressed in either one can be mapped to an equivalent inference or computation in the other.

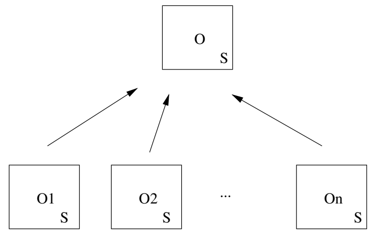

Figure 3: Merge

4.4 Our View

In the merge process we have, on one hand, a set of ontologies (at least two) that are going to be *merged* ($O_1, O_2, ..., O_n$, Figure 3), and on the other hand, the *resulting* ontology (O, Figure 3). The goal is to make a more general ontology about a subject by gathering into a coherent bulk, knowledge from several other ontologies in that same subject. The subject of both the merged and the resulting ontologies are the same (S, Figure 3) although some ontologies are more general than others, that is, the level of generality of the merged ontologies may not be the same.

In merge it may be difficult to identify regions in the resulting ontology that were taken from the merged ontologies and that were left more or less unchanged, specially in the cases of unification. In the case of unification, knowledge from the merged ontologies is homogenized and altered through the influence of one source ontology on another (is spite of the fact that the source ontologies do influence the knowledge represented in the resulting ontology). In other cases the knowledge from one particular source ontology is scattered and mingled with the knowledge that comes from the other sources. In the cases of alignment where minimal changes are required it may be possible to identify some regions in the resulting ontology that were taken from the merged ontologies. One can certainly find the concepts from the source ontologies unchanged since no changes are made either to axioms, definitions, proofs, or computations in any of the source ontologies. In the hypothetical example presented in Figure 4 we show a possible merge process of ontologies O_1 and O_2 into the resulting ontology O. In this example, concepts from the source ontologies can be identified.

The way the merge process is performed is still very unclear. So far, it is more of an art. Several different approaches to the problem have been put forward by different groups addressing the issue in different domains: in the natural language domain it was done by hand; in the medical domain a general methodology was proposed; and in search for generalized upper-models a group iterative approach to reach consensus was proposed and a one lonely researcher

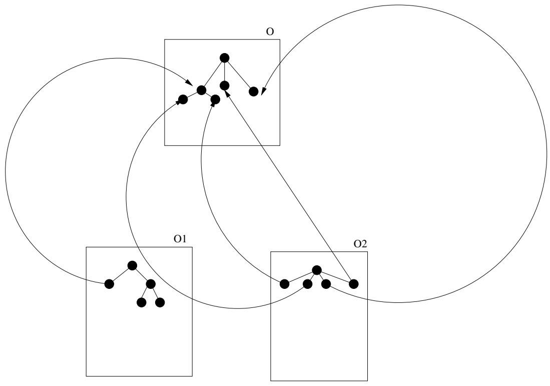

Figure 4: The resulting ontology in a merge process

effort based on philosophy was followed. So, one of the problems of merge is what methodologies should be used to do it. There are a few different methodologies but there is no consensus on the methodology to follow to merge ontologies either on the same or in different domains. It even is not clear whether there could be a domain independent methodology. The common trace in the merging process is that, at least, an initial group of ontologies (more than one) is analyzed together in its initial steps. While ONIONS starts with all ontologies at the beginning of the merge process, Skuce begins with a selected initial group of ontologies that is incrementally enlarged. While in integration we could repeat the process for several integrated ontologies, one ontology at a time, in the merge process an initial set of ontologies is, at least, analyzed together in the initial steps.

In merge there are no operations and what is mainly used are abstraction, matching and generalization capabilities and common sense. A deep knowledge in Philosophy also seems to help in this rather undefined process. Issues concerning generalized upper levels have been studied for more than 2000 years in that science. Some general guidelines to guide the abstraction process, like ONIONS, seem to help this rather complex process. The problem seems to be to find them.

5 Use/Application

In this section we present some ontologies that were built either from scratch, through integration, or through merge that were actually used by real applications. We also present work aiming at easing and characterizing use. We should stress that there aren't many reports of application of ontologies in the literature and the few existing reports do not give enough technical details on how the ontology was used by the application.

5.1 Easing Use

In (Uschold & Gruninger 1996) a series of uses of ontologies was identified: *communication* "between people with different needs and viewpoints arising from their particular contexts"; *inter-operability* among "different users that need to change data and who are using different software tools"; and *systems engineering* related to "the role ontologies play in the operation of software systems".

In (Uschold 1998) a set of ten features is proposed to classify and characterize ontology applications. Once the characterization of applications is made, people new in the area wanting to build an ontology application could look up that information and avoid re-inventing the wheel[12]. Also to ease the use of ontologies, (V. et al. 1998) presents a taxonomy of seventy features and a WWW-broker that help future users to select the most adequate and suitable ontology for the application they have in mind. Another problem related to use is the integration of Problem Solving Methods with ontologies (Chandrasekaran, Josepheson & Benjamins 1998).

5.2 Ontologies To Be Used

The ontology developed by ARPA/Rome Laboratory Planning Initiative (Lehrer 1993), for representing plans and planning information is composed of: (1) an abstract ontology setting the more general categories (such as space, time, agents); (2) a set of modular specialized ontologies which enlarges the general categories with sets of concepts and alternative theories of more detailed notions commonly used by planning systems (for instance, specific ontologies of temporal relations). The specialized ontologies also provide definitions of concepts when several alternative sets of concepts are commonly used to describe the same subject in the abstract ontology. These ontologies (abstract and concrete) are used by disparate and communicating agents. This ontology can be classified as an inter-operability one, according to the framework of uses in (Uschold & Gruninger 1996).

Among the various ontologies already built for use we can refer: CYC (Lenat & Guha 1990, Lenat & Guha 1994, Lenat, Guha, Pittman, Pratt & Shepherd 1990)[13], GUM (Bateman, Magnini & Fabris 1995), PIF (Lee, Gruninger, Jin, Malone, Tate, Yost & other members of the PIF Working Group 1996) the GALEN project, UMLS.

5.3 Tools For Use

KACTUS[14] (Knowledge About Complex Technical systems for multiple USe) project (Schreiber et al. 1995, Laresgoiti, Anjewierden, Bernaras, Corera, Schreiber & Wielinga 1996) aims at "finding and building methods and tools to enable reuse and sharing of technical knowledge". It began by modeling reasoning processes and then went on

[12] For instance, look up the techniques used to build similar applications.

[13] `http://www.cyc.com`

[14] `http: //swi.psy.uva.nl/projects/NewKACTUS/home.html`

modeling ontologies. KACTUS developed a methodology, a tool and a library to help the construction of KBS for complex technical domains. According to the framework of uses in (Uschold & Gruninger 1996), the ontologies kept at the library can be classified as systems engineering ones.

5.4 Ontologies That Were (Re)Used

In (Laresgoiti et al. 1996), a case of reuse of a previously built ontology in a new application is presented. The conclusions were that the reuse of ontologies saved a lot of effort in implementing applications. However reuse is almost never complete so "reusing any ontology for different purposes than those for which the ontology was built will always require modifications or tuning for the new purposes". They also concluded that a clear organization and a clear methodology is needed so that people unfamiliar with the ontology are able to use it.

Another important application that uses ontologies is described in (Bernaras et al. 1996). First, an ontology for diagnosis in electrical transmission networks was built. Then, an ontology for service recovery planning on the same domain was built. Finally, they were both unified in the sense of merge&integration. From the unified ontology several applications that needed knowledge about electrical power transmission systems were developed. The conclusions were that modularization and hierarchical organization seem to be good ontology structuring and design principles and that abstraction and standardization, although good principles, should be used with care. Concrete objects were more usable than more abstract ones (although abstraction is a basic principle to reusability). The fact is that to implement specific concepts from more generic ones demanded a big design effort, more than implementing those concepts from other specific concepts from related applications. The conclusions reached in (Bernaras et al. 1996) about the problem of ontology use are that these ontologies should be built only a small level up in generality than the one used for a specific application, so that the implementation of the ontology in other applications won't involve a big design effort.

EngMath was reused to build an application for aircraft design (Uschold, Healy, Williamson, Clark & Woods 1998). The process of reusing the ontology can be summarized as: (1) understanding the ontology and finding the kernel of reusable knowledge; (2) translate the ontology (that was initially written in Ontolingua) into Slang (Waldinger, Srinivas, Goldberg & Jullig 1996); (3) specify and refine the task definitions in an iterative process moving closer and closer to the implementation of that specification; (4) verify each refinement step; (5) integrate the resulting specification in the specification of the application and refine its result into executable code. The conclusions reached were that it actually was cost-effective to reuse the ontology instead of building it from scratch; and that the translation of ontologies involves a significant manual effort (which is considerable difficult to automatize).

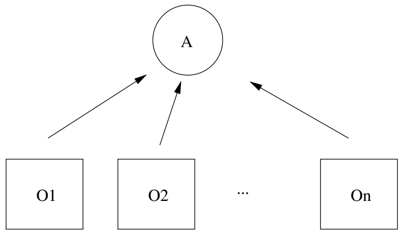

Figure 5: Use

Other ontologies that were actually used to build applications and whose application is discussed in the literature are PhySys (Borst 1997), the Enterprise Ontology (Uschold, King, Moralee & Zorgios 1998), the Reference Ontology (V. et al. 1998), among others.

5.5 Our View

In use, there are one or more ontologies involved ($O_1, O_2, ..., O_n$, Figure 5) and there is no resulting ontology. (A, in Figure 5, is the application using the ontology). One cannot draw any conclusions as to the architecture of the resulting application because that depends on the application itself. In the case that several ontologies are used they should be compatible among themselves. There are several issues involved when analyzing compatibility: language, ontological commitments, level of detail, context, etc. However, we think that there is an order of importance among those different compatibility criteria. If two ontologies are not compatible in their ontological commitments then all other criteria are irrelevant and it is meaningless to analyze them. The ontologies should also satisfy a set of characteristics such as level of generality, modularity, etc., as discussed in (Bernaras et al. 1996). Finally, only verified (Gómez-Pérez et al. 1995) ontologies should be considered in use.

So far, there are no operations identified in the literature. The ontologies used in applications should probably have a set of configurable parameters. In each application developed based on that ontology those parameters should be customized. In what concerns use's methodological aspects a lot of work needs to be done. One can probably find some guidelines that can ease the process of using the ontology, as (Laresgoiti et al. 1996) tries to establish. We think that a set of specific methodologies based on the kind of application involved could also help the use process. Among these specific methodologies are methodologies for use in KBS, Internet brokers, etc.

6 Conclusions

In this article we clarify the meaning of the word integration in the OE field. The three acknowledged meanings associated to the word "Integration" should actually be defined using the following words:

1. Integration - In the case of building a new ontology reusing (by composing) other available ontologies.

2. Merge - In the case of building an ontology unifying knowledge of several ontologies into a single one.

3. Use/Application - In the case of integrating ontologies in applications.

Both integration and merge are processes that aim at building ontologies from other ontologies. However this is where similarities end. These processes are quite different one from the other, as we have discussed. Use/Application is a completely different process. The objective is not to build an ontology. The aim is to build an application using ontologies.

In this article we present a series of partial conclusions on the three different processes that we have identified. We have identified the main characteristics of the integration process, presented some of the operations proposed in the literature to do integration and presented how current ontology building methodologies deal with this issue. Further work is needed in order to identify a larger set of operations that can be used in the integration process, specify how those operations should work in each phase of the ontology development life cycle and specify the overall integration process. Further research in integration is within our plans.

We have also identified the main characteristics of the merge process. In the literature, no operations to do merge are proposed and there are very few methodologies. Some work is needed in order to specify more and better defined methodologies for this process of building ontologies and see whether these methodologies can be applied to different subjects, domains and contexts.

Finally, we have identified the main characteristics of the use process. As in the previous process, no operations to do use were proposed although there are some guidelines in the literature about the features that those ontologies should have. Some work is needed in order to develop methodologies to do use of applications in spite of the fact that these methodologies will be application dependent.

7 Acknowledgments

We thank all members of GIA at Instituto Superior Técnico for their support, in particular Ana Cachopo, João Cachopo and António Leitão. We thank all members of the Knowledge Sharing and Reuse group at Lab. de Inteligencia Artificial at UPM for their support, in particular Mariano Fernández.

References

Bateman, J. A., Kasper, R. T., Moore, J. D. & Whitney, R. A. (1989), A General Organization of Knowledge for Natural Language Processing: The PENMAN Upper Model. research report, USC/Information Sciences Institute, Marina del Rey.

Bateman, J., Magnini, B. & Fabris, G. (1995), The Generalized Upper Model Knowledge Base: Organization and Use, *in* N. Mars, ed., 'Towards Very Large Knowledge Bases', IOS Press, pp. 60–72.

Benjamins, R. & Fensel, D. (1998), The Ontological Engineering Initiative $(KA)^2$, *in* N. Guarino, ed., 'Formal Ontology in Information Systems', IOS Press, pp. 287–301.

Bernaras, A., Laresgoiti, I. & Corera, J. (1996), Building and Reusing Ontologies for Electrical Network Applications, *in* W. Wahlster, ed., 'ECAI96 Proceedings', John Wiley&Sons, pp. 298–302.

Blázquez, M., Fernández, M., García-Pinar, J. M. & Gómez-Pérez, A. (1998), Building Ontologies at the Knowledge Level Using the Ontology Design Envirnoment, *in* 'Knowledge Acquisition Workshop'.

Borst, P. (1997), Construction of Engineering Ontologies for Knowledge Sharing and Reuse, PhD thesis, Tweente University.

Borst, P., Akkermans, H. & Top, J. (1997), 'Engineering Ontologies', *International Journal of Human Computer Studies special issue "Using Explicit Ontologies in KBS Development"* **46**(2/3), 365–406.

Borst, P., Benjamin, J., Wielinga, B. & Akkermans, H. (1996), An Application of Ontology Construction, *in* 'ECAI96's workshop on Ontological Engineering'.

Chandrasekaran, B., Josepheson, J. R. & Benjamins, V. R. (1998), Ontology of Tasks and Methods, *in* 'ECAI98's workshop on Application of Ontologies and Problem Solving Methods'.

Cote, R. A., Rothwell, D. J. & Brochu, L., eds (1994), *SNOMED International*, 3rd edn, Northfield III: College of American Pathologists.

Dalianis, H. & Persson, F. (1997), Reuse of an Ontology in an Electrical Network Domain, *in* 'AAAI97 Spring Symposium Series, workshop on Ontological Engineering'.

Farquhar, A., Fikes, R., Pratt, W. & Rice, J. (1995), Collaborative Ontology Construction for Information Integration, Technical Report KSL-95-63, Knowledge Systems Laboratory, Stanford University.

Farquhar, A., Fikes, R. & Rice, J. (1996), The Ontolingua Server: A Tool for Collaborative Ontology Construction, *in* 'Knowledge Acquisition Workshop'.

Farquhar, A., Fikes, R. & Rice, J. (1997), Tools for Assembling Modular Ontologies in Ontolingua, *in* 'AAAI97 Proceedings', pp. 436–441.

Fernández, M. (1996), CHEMICALS: Ontologia de Elementos Químicos. Proyecto Fin de Carrera, Fac. de Informática, UPM.

Fernández, M., Gómez-Pérez, A. & Juristo, N. (1997), METHONTOLOGY: From Ontological Art Towards Ontological Engineering, *in* 'AAAI97 Spring Symposium Series, workshop on Ontological Engineering'.

Fernández, M., Gómez-Pérez, A., Sierra, A. P. & Sierra, J. P. (1999), 'Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment', *IEEE Intelligent Systems* **14**(1), 37–46.

Gabrieli, E. A. (1989), 'A New Electronic Medical Nomenclature', *Journal of Medical Sciences* **3**(6).

Gangemi, A., Pisanelli, D. M. & Steve, G. (1998), Ontology Integration: Experiences with Medical Terminologies, *in* N. Guarino, ed., 'Formal Ontology in Information Systems', IOS Press, pp. 163–178.

Gangemi, A., Steve, G. & Giacomelli, F. (1996), ONIONS: an ontological methodology for taxonomic knowledge integration, *in* 'ECAI96's workshop on Ontological Engineering'.

Gómez-Pérez, A., Juristo, N. & Pazos, J. (1995), Evaluation and Assessement of the Knowledge Sharing Technology, *in* N. Mars, ed., 'Towards Very Large Knowledge Bases', IOS Press, pp. 289–296.

Gruber, T. (1993), Ontolingua: A Mechanism to Support Portable Ontologies, Technical Report KSL-91-66, Knowledge Systems Laboratory, Stanford University.

Gruber, T. (1995), 'Towards Principles for the Design of Ontologies for Knowledge Sharing', *International Journal of Human Computer Studies* **43**(5/6), 907–928.

Gruber, T. & Olsen, G. R. (1994), An Ontology for Engineering Mathematics, *in* E. S. J. Doyle & P. Torasso, eds, 'KR94 Proceedings', Morgan Kaufmann, pp. 258–269.

Gruninger, M. (1996), Designing and Evaluating Generic Ontologies, *in* 'ECAI96's workshop on Ontological Engineering'.

Guarino, N. (1997), Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration, *in* M. T. Pazienza, ed., 'Information Extraction: A Mutidisciplinary Approach to an Emerging Information Technology', Springer Verlag.

Guarino, N. (1998), Formal Ontology and Information Systems, *in* N. Guarino, ed., 'Formal Ontology in Information Systems', IOS Press, pp. 3–15.

Guarino, N., Carrara, M. & Giaretta, P. (1994), Meta-Level Categories, *in* E. S. J. Doyle & P. Torasso, eds, 'KR94 Proceedings', Morgan Kaufmann, pp. 270–279.

Hovy, E. (1996), What Would it Mean to Measure an Ontology? Work in progress. Unpublished ms, USC/ISI.

Humphreys, B. L. & Lindberg, D. A. B. (1992), The Unified Medical Language Systems Project, *in* K. C. Lun, ed., 'MEDINFO92', Elsevier Science Publishers.

Humphreys, B. L. & Lindberg, D. A. B. (1993), 'The UMLS Project: Making the Conceptual Connection Between Users and the Information They Need', *Bulletin of the Medical Library Association* **81**(2).

Knight, K., Chander, I., Haines, M., Hatzivassiloglou, V., Hovy, E. H., Iida, M., Luk, S. K., Withney, R. A. & Yamada, K. (1995), Filling Knowledge Gaps in a Broad-Coverage MT System, *in* 'IJCAI95 Proceedings', pp. 1390–1397.

Knight, K. & Luk, S. (1994), Building a Large Knowledge Base for Machine Translation, *in* 'AAAI94 Proceedings', pp. 773–778.

Laresgoiti, I., Anjewierden, A., Bernaras, A., Corera, J., Schreiber, A. T. & Wielinga, B. J. (1996), Ontologies as Vehicles for Reuse: a mini-experiment, *in* 'Kowledge Acquisition Workshop'.

Lee, J., Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost, G. & other members of the PIF Working Group (1996), Process Interchange Format (PIF) for Sharing Ontologies, *in* 'ECAI96's workshop on Ontological Engineering'.

Lehrer, N. (1993), KRSL Specification Language, Technical Report 2.0.2, ISX Corporation. Also available as `http://isx.com/pub/ARPI/ARPI-pub/krsl/krsl-info.html`.

Lenat, D. & Guha, R. (1990), *Building Large Knowledge-Based Systems, Representation and Inference in the CYC Project*, Addison Wesley.

Lenat, D. & Guha, R. (1994), 'Enabling Agents to Work Together', *Communications of the ACM* **37 (7)**, 127–142.

Lenat, D., Guha, R., Pittman, K., Pratt, D. & Shepherd, M. (1990), 'CYC: Toward Programs with Common Sense', *Communications of the ACM* **33** **(8)**, 30–49.

Miller, G. (1990), 'WordNet', *International Journal of Lexicography* **3**.

Newell, A. (1982), 'The Knowledge Level', *Artificial Intelligence* **18**, 87–127.

Rector, A. L., Gangemi, A., Galeazzi, E., Glowinski, A. & Rossi-Mori, A. (1994), The GALEN CORE Model Schemata for Anatomy: Towards a Re-Usable Application Independent Model of Medical Concepts, *in* 'Medical Informatics Europe'.

Rector, A. L., Solomon, W. D. & Nowlan, W. A. (1995), 'A Terminology Server for Medical Language and Medical Information Systems', *Methods of Information in Medicine* **34**, 147–157.

Schreiber, G., Weilinga, B. & Jansweijer, W. (1995), The Kactus View on the 'o' World, *in* 'IJCAI95's workshop on Basic Ontological Issues in Knowledge Sharing'.

Skuce, D. (1997), How We Might Reach Agreement on Shared Ontologies: A Fundamental Approach, *in* 'AAAI97 Spring Symposium Series, workshop on Ontological Engineering'.

Sowa, J. F. (1995), 'Top-Level Ontological Categories', *International Journal of Human Computer Studies* **43**(5/6), 669–685.

Sowa, J. F. (1997), Electronic communication in the `onto-std` mailing list, 4th of December.

Steve, G. & Gangemi, A. (1996), ONIONS Methodology and the Ontological Commitment of Medical Ontology ON8.5, *in* 'Kowledge Acquisition Workshop'.

Swartout, B., Patil, R., Knight, K. & Russ, T. (1997), Toward Distributed Use of Large-Scale Ontologies, *in* 'AAAI97 Spring Symposium Series, workshop on Ontological Engineering'.

Uschold, M. (1998), Where are the Killer Apps?, *in* 'ECAI98's workshop on Application of Ontologies and Problem Solving Methods'.

Uschold, M. & Gruninger, M. (1996), 'Ontologies: Principles, Methods and Applications', *Knowledge Engineering Review* **11**(2).

Uschold, M., Healy, M., Williamson, K., Clark, P. & Woods, S. (1998), Ontology Reuse and Application, *in* N. Guarino, ed., 'Formal Ontology in Information Systems', IOS Press, pp. 179–192.

Uschold, M. & King, M. (1995), Towards a Methodology for Building Ontologies, *in* 'IJCAI95's workshop on Basic Ontological Issues in Knowledge Sharing'.

Uschold, M., King, M., Moralee, S. & Zorgios, Y. (1998), The Enterprise Ontology, Technical Report AIAI-TR-195, Artificial Intelligence Applications Institute, University of Edinburgh. To appear in the Knowledge Engineering Review, vol 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate).

V., J. A., Gomez-Perez, A., T., A. L. & Pinto, H. S. (1998), $(\mathrm{ONTO})^2$Agent: An Ontology-Based WWW Broker to Select Ontologies, *in* 'ECAI98's workshop on Application of Ontologies and Problem Solving Methods'.

van Heist, G., Schreiber, A. T. & Wielinga, B. J. (1997), 'Using Explicit Ontologies in KBS Development', *Internacional Journal of Human-Computer Studies* **45**, 183–292.

Waldinger, R., Srinivas, Y. V., Goldberg, A. & Jullig, R. (1996), Specware Language Manual. -.

WHO (1994), *Classification of Diseases*, 10th edn, WHO.