

ACQUISITION AND STRUCTURING OF AN ONTOLOGY WITHIN CONCEPTUAL GRAPHS*

J. BOUAUD, B. BACHIMONT, J. CHARLET and P. ZWEIGENBAUM
DIAM – Dép. de Biomathématiques CHU Paris 6
É Service d'Informatique Médicale AP-HP
91, boulevard de l'Hôpital, F-75634 Paris CEDEX 13, France

ABSTRACT

The elicitation of the ontology – *i.e.* the objects of a domain – is a key issue of conceptual modelling and therefore of knowledge acquisition. The Conceptual Graph Theory provides a knowledge representation formalism to be used in knowledge-based systems with an explicit “type lattice” to account for the ontology. Since knowledge is in most AI applications non formal, it has to be normalized to ensure that the formal exploitation of its representation conforms to its meaning in the domain. Noting the intensional nature of types, which reflect the essences of the objects they denote, this normalization relies on a commitment on type definitions by necessary and sufficient conditions at the knowledge level. Our claim is that the taxonomic structure that accounts for the intensional nature of the ontology can be nothing but a tree, precluding tangled taxonomies. From this starting point, we derive methodological principles to constrain the acquisition of the “type tree”, thus helping in the design of a domain ontology. These principles are currently applied to acquire the ontology and related knowledge in the context of the knowledge-based part of MENELAS, a natural language understanding project in the medical domain, which uses Conceptual Graphs as its core formalism.

1. Introduction

Taxonomies and hierarchies are widely used in knowledge-based systems, especially in the medical domain^{1,2}. Most of current knowledge representation formalisms indeed give a central place to IS-A hierarchies^{3,4}. These hierarchies enumerate the categories of objects that are considered to exist in a given domain, and structure them through subsumption. Those categories constitute the *ontology* of the domain.

The ontology is the heart of any knowledge description: knowledge is intimately related to the ontology, since it is necessarily expressed in terms of this ontology. Therefore, designing the ontology of a domain is a key issue in the knowledge acquisition process^{5,6,7} during the step of knowledge elicitation⁸. When developing an

*This work has been mainly supported by European Community in the framework of the AIM program (Advanced Informatics in Medicine) in project # 2023: “MENELAS, *an access system for medical records using natural language.*”

ontology, one is faced with numerous problems and representational alternatives. One has to make choices, or “ontological commitments,”⁹ for which very few guidelines exist in the literature^{10,11}.

In this paper, we present methodological principles to control the acquisition of an ontology in the context of knowledge-based systems. This work results from our experience in MENELAS, a natural language understanding system based on Conceptual Graphs³. The objective of MENELAS (European project AIM A2023) is to allow users to access information contained in medical reports by performing various interesting tasks for them: generation of nomenclature codes, content-based information retrieval, etc.

MENELAS relies upon the general hypothesis that understanding consists in building a language-independent conceptual representation of the world situation described in the text¹²; it focuses on patient discharge summaries in French, English and Dutch in the domain of cardiac catheterisation. The overall principle of the system is to analyze the contents of medical reports and to store them in a database as a set of conceptual structures which represent their “meaning”; these structures may be seen as the core representation of the narratives. They are then the basis for the identification of relevant nomenclature codes, and may be consulted to retrieve specific information contained in a text¹³. They should as much as possible include information that was implicit in the texts and is evident for the target readers¹⁴.

To solve this last point, MENELAS adopts a knowledge-based approach to natural language understanding, and relies on a large body of medical and common-sense knowledge. All semantic and domain knowledge, as well as the representation of text meaning, are expressed in Conceptual Graphs (CGs). Indeed, one of the motivations in the design of the CG model³ was to capture the semantics of natural language; besides, CGs have already been used in medical systems^{15,16,17}. We focus here on the knowledge-based part of the system (pragmatics), and not on its linguistic aspects (morphology, syntax, and semantics). Our concern is to represent knowledge and data with CGs and reason with them. As a knowledge representation formalism, CGs constrain the way knowledge can be represented. The *conceptual support* is made of *types* and *relations* organized in taxonomies. These structures are the keystone of knowledge representation since they provide the basic conceptual vocabulary that all domain descriptions will be built of. Moreover, they determine all valid operations on those representations. While types are hierarchically organized in an IS-A *lattice*, relations are organized in an IS-A *tree*. This conceptual support is a good candidate to be a repository of the domain ontology.

Setting up the support is a preliminary task of building a knowledge based application using CGs. However, medicine is a typical domain where knowledge is by nature descriptive, mainly expressed in natural language, and not formalized. Though a technical and restricted domain, concepts do not have clear and precise definitions.

Since a key issue in representing knowledge is to ensure that its formal exploitation at the symbol level by a program conforms to its meaning at the knowledge level for people¹⁸, descriptive, informal, knowledge has to be normalized. A commitment on the meaning of concepts is needed. Noting the intensional nature of types on the one hand, and the need for type definitions by necessary and sufficient conditions in order to be able to use a taxonomy on the other hand, we argue that this commitment has to take place at the knowledge level and is tied up to the essence of objects, namely the ontology. Assuming that types are defined through their essence, our claim is that the taxonomic structure to account for the intensional nature of the ontology can be nothing but a tree, precluding tangled taxonomies.

From this starting point, we derive methodological principles to constrain the acquisition of the “type tree”, thus helping in the formalization of a domain ontology. These principles are currently applied to acquire the domain ontology and related knowledge in the context of MENELAS.

The paper is organized as follows. Section 2 starts on the problems related to the representation and acquisition of type hierarchies. An overview of conceptual graphs is given in section 3. The role of types and the way they are used are mainly studied as they are fundamental for knowledge representation. Section 4 contains the rationale for our claim that the taxonomic structure of the ontology is a tree. Derived methodological principles for structuring ontological types in a CG knowledge based application are presented in section 5. A discussion about these ontology acquisition principles takes place in section 6. The last section sums up this work.

2. Background

2.1. What’s a Type Hierarchy?

A general definition for a *conceptual type*, or simply type, is that of a “category of objects”.^{*} Types are organized in a hierarchy[†] according to their levels of generality, often called *taxonomy*¹⁹. The taxonomy is used for storing information with a type and making it automatically available to more specific types by means of a mechanism of inheritance. More general types are said to *subsume* more specific types. The subsumption relation over types is the transitive closure of the basic taxonomic links. An important notion is that individuals belonging to a type also belong to its subsumers. Types may simply be data structure specifications. But, within the framework of knowledge representation, types aim at representing the “categories of thought”, and constitute the *conceptual basis*³.

^{*}Many authors use the term *concept* for a conceptual category. In the conceptual graph terminology, this notion is termed a *type* while a “concept” describes an occurrence of a type (see 3.1). However, in the rest of the paper, we shall use either *type* or *concept* for a conceptual category, or a notion.

[†]The term *hierarchy* is used here for any partial ordering, but not specifically for a *tree*.

2.2. Acquiring a Type Hierarchy

Building a taxonomy that represents the types of object on which a knowledge-based system is supposed to reason is a knowledge acquisition issue. Considering this issue, Lehmann²⁰ reports three potential sources for a taxonomy: (i) the hierarchy is given by a designer who supplies the taxonomic links by hand between atomic types; (ii) the hierarchy is induced by some other formal structure attached to types; and (iii) the hierarchy may emerge directly from a set of data. However, real problems arise with the way a taxonomy is used, the way the taxonomic links are interpreted, either by people or computers. Considering conceptual taxonomies, Woods¹⁹ introduces a distinction among five distinct senses for subsumption. Since types represent the categories of thought³, understanding the very nature of the subsumption represented in a taxonomy is fundamental.

Thus, determining how a taxonomy accounts for knowledge is the issue. In the first mentioned case, we cannot rely on but the designer's skills and assume s/he is able to describe and organize correctly his/her knowledge of the domain. In the second case, the question is to know to which extent the structural properties that rule subsumption account for knowledge. In the third case, which is some kind of automatic clustering, the interpretation, or characterization of the meaning, of the emerging categories remains a problem. Finally, another fourth solution remains possible: reuse some previously built taxonomy in the same domain.

2.2.1. Building Taxonomies

In the medical domain, taxonomies have been developed for a long time now for classifications²¹ and nomenclatures²². Faced with the proliferation of differing vocabularies and conceptualizations, unifying projects have been launched. The Unified Medical Language System (UMLS²³) aims at providing a bridge across the existing schemes, by compiling and cross-linking all underlying concepts (the "Meta-thesaurus"). A central piece of knowledge in the UMLS is the semantic network¹, whose backbone is a hierarchy of "semantic types." The GALEN project²⁴ aims at designing a generative description and structuring of medical concepts, and relies on the use of a subsumption language.

In another but larger domain, CYC⁵ is a huge knowledge representation enterprise which aims at providing common-sense knowledge to knowledge based systems. The central piece of the project is an ontology of the categories and things in the world. The design approach of the authors is largely empirical and driven by examples even if they keep their work supported by theoretical foundations.

Although they did not build huge taxonomies, Brachman *et al.*¹¹, in the context of description languages, give useful guidelines of a simple engineering methodology for CLASSIC. In such a system in which subsumption is structural, a particular attention is paid to the elaboration of explicit type definitions by means of essential properties.

In the context of CGs, though the conceptual basis is a taxonomy³, very few

guidelines are given for its construction. However, Tepfenhart²⁵ proposes constraining principles for the constitution of a conceptual basis using the Situation Data Model. Attempts to control interactively the taxonomy structure have been explored^{26,27} for both atomic and structured types. Such a control relies mainly on formal structural properties of either the descriptions or the taxonomy itself, and the designer is required to solve conflicts at the knowledge level.

2.2.2. Reusability of Ontologies

The reusability of ontologies is the foundation of numerous projects. CYC is one of the most famous. The ontology is the central part of the CYC knowledge base. For the authors this ontology is rather universal than reusable in so far as it carves up the universe to represent enough common-sense and encyclopedic knowledge to support natural language capabilities. KIF²⁸ is a logic-based, implementation independent, language for knowledge representation. It provides for the definition of objects of a conceptualization and therefore the description of an ontology. Assuming these ontologies are reusable, Gruber's Ontolingua⁹ makes these KIF ontologies portable since it translates them into several knowledge representation system idioms, thus enabling knowledge sharing.

In medicine, our attempt to reuse UMLS in MENELAS²⁹ has suffered from some mismatches. The semantic hierarchy of UMLS where only IS-A relations have been kept, seemed a good prototype for the ontology of MENELAS. But some problems appeared and we had to take some distance from this initial structure. On the one hand, our purpose is different from that of UMLS, and we had to cater for general concepts that are useful for natural language understanding. On the other hand, we focus on the domain of cardiology, and we had to enrich and refine some of the lower parts of the hierarchy. Finally, the task for which an application is developed fixes a peculiar point of view on the ontology, and the reusability of this ontology for another system, *i.e.* another task, seems difficult³⁰.

As pointed out in section 2.2, most of the problems that arise when one tries use an existing taxonomy deal with the semantics of subsumption. Woods¹⁹ reports for instance that "Many users of KL-ONE used links in ways that violated the semantics intended by its designers." and argues for a clear semantics for taxonomic links.

3. Types and Conceptual Graphs

3.1. *The Conceptual Graph Theory*

The Conceptual Graph theory has been proposed by Sowa³ in the context of semantic networks as a unifying attempt to capture the meaning of natural language. He proposed a graphical formalism intended to serve as an intermediate representation between natural language and logic³¹. We only give a brief overview of the fundamental notions of the basic CG formalism, and then insist on the type notion.

CGs are intended to account for statements about the world in a conceptual representation; an assertion such as *Mary is admitted for myocardial infarction* could be represented by the following CG:

[PATIENT:Mary]<--(PAT)--[HOSPITALIZATION]--(MOTIV)-->[MYOCARDIAL-INFARCTION]

On a formal point of view³², a conceptual graph is a bipartite, connex, finite graph with two kinds of nodes: concept-nodes, or *concepts*, and relation-nodes or *links*. Every node has a symbolic label. A link label is made of a *relation* and a concept label is made of two sub-labels: a *type* and a *marker* which correspond respectively to the concept type and the concept referent. Concepts are graphically represented by rectangles and links by circles and arrows. In the linear notation, concepts are delimited by square brackets and links by parentheses. In the previous example, PATIENT, HOSPITALIZATION, and MYOCARDIAL-INFARCTION are types, PAT and MOTIV, relations. Mary is an individual marker.

Building a CG depends on the possible labels that can be used, *i.e.* on the sets of types, relations, and markers. These sets constitute the *support* of a CG-based representation. All properties of, and operations on CGs depend on their own structure and on the support they share.

3.1.1. The Support: Types, Relations, and Markers

The set of types is structured by a partial order (\leq), forming a hierarchy which must be a lattice in the mathematical sense, and is therefore called the *concept type lattice* (CTL). For each pair of types, there must be a unique least upper bound and a unique greatest lower bound. The universal type is noted \top and the absurd type \perp . There is one and only one kind of taxonomic link between types.

Relations are distinguished from types. The set of relations has in the basic framework no particular structure but could have the same structure as the CTL. In MENELAS, it is structured as a tree. Moreover, each relation has a signature that specifies its arity and the types of its arguments.

Markers are made of a set of *individual markers* and a *generic referent*, noted $*$. The latter subsumes all the formers. A conformity relation relates types to the individual markers that conform to them. When an individual marker conforms to a type, then it conforms to all its supertypes. In practical applications with CGs, there are no predefined markers, but they are created on demand according to the application needs.

3.1.2. Graph Subsumption and CG Operations

CGs can be compared according to their structure using the subgraph morphism properties. Moreover, the subsumption relations that exist over the node labels provides additional ways to compare nodes. The *CG projection* corresponds to a subgraph morphism that may restrict node labels. Projection induces a subsumption

relation over CGs: if there exists a projection between two CGs, the first subsumes the second. A set of CGs is consequently implicitly structured by a generalization/specialization relation by means of projection. Methods and algorithms have been studied and designed to test subsumption between two CGs³³ and allow classification^{34,35}.

Specialization operators are defined; when applied to CGs, they yield a new CG which is more specific than its antecedents. Main specialization operators are *restriction* and *join*. The first one is basically used to restrict one node label to one of its sub-labels, the second one to merge nodes from different CGs. In the latter case, the lattice structure of labels is important to ensure a deterministic merging of nodes since their greatest lower bound is used to tag the image node.

3.1.3. Logical Interpretation

CGs are also another notation for logic. The ϕ function maps a CG into a well formed formula of first order predicate logic. In this framework, types are turned into monadic predicates, relations into n-adic predicates, individual markers into constants, and generic markers into variables. Applying ϕ to a CG yields a wff. Considering the simple CG we described above, the logical formula associated with it is:

$$\begin{aligned} \exists x \exists y \text{ patient}(\text{Mary}) \wedge \text{pat}(x, \text{Mary}) \wedge \text{hospitalization}(x) \\ \wedge \text{motiv}(x, y) \wedge \text{myocardial infarction}(y) \end{aligned}$$

The type hierarchy is turned into logical implications. For all pairs of types (t, t') , if $t' \leq t$ then $\forall x t'(x) \supset t(x)$. It has been shown that the same holds for CG subsumption. If a graph G' is a specialization of G then $\phi(G') \supset \phi(G)$. Thus, type subsumption and graph subsumption have basically the same logical interpretation.

3.2. Defining Types

On the one hand, CGs are given a formal, logical, interpretation, and on the other hand, they are supposed to describe a conceptual situation of mind. We address here the role types intend to play in the CG formalism.[‡]

3.2.1. Definition by Genus and Differentiae

Considering that types are used to represent “categories of objects” of a domain, a common way to define a new category since Aristotle is to provide it with a *definition* specified by *genus* and *differentiae*. The genus is another type, subsuming the new one, and the differentiae are some additional properties that can be supported by the genus. The conditions imposed on the genus and differentiae must be at least *necessary* to the new type. When these conditions are also *sufficient* to determine

[‡]In the rest of the paper we shall only focus on types. Relations to some extent are equivalent to types, since they can be explicitly defined in terms of types.

the new type then the definition is said to be *complete*. Otherwise, they are only necessary and the definition is said to be *partial*.

3.2.2. Conceptual Graph Type Definition

Considering a type to be defined as a subtype of its genus, and assuming that the differentiae can be expressed in terms of a CG, then the CG formalism provides means to define such a type by stating explicitly the differentiae. CGs are then used as a type forming language like description languages such as KL-ONE⁴. For instance, the type INFARCTED-PATIENT can be defined as a patient who has been admitted for a myocardial infarction. Assuming the representation of a *patient admitted for myocardial infarction* can be expressed as a CG, then the type definition would be the following:

```
Type INFARCTED-PATIENT(*x) is
  [PATIENT:*x]<--(PAT)--[HOSPITALIZATION]--(MOTIV)-->[MYOCARDIAL-INFARCTION]
```

Then INFARCTED-PATIENT becomes a subtype of PATIENT. The genus is *patient* and the differentiae is *admitted for a myocardial infarction*, and the graph that represents them is called the body of the definition. The genus is derived from the concept of the body pointed by the formal parameter of the definition.

Now, let us consider a type T defined explicitly by its genus G and differentiae D, then T implies G and D, enabling G and D to be satisfied when T has been asserted. This corresponds to the minimal CG type expansion. If T was completely defined, then the implication between T and its definition, G and D, would become an equivalence, enabling the assertion of T when G and D are satisfied. This corresponds to the CG type contraction.

The subsumption hierarchy of CGs also applies to the bodies of type definitions. Defined types are consequently partially ordered and constitute part of the CTL. An important point is that the properties that define types, be they explicit or not, are inherited by their subtypes, in the sense that they are logically implied. This is the basic inference mechanism of inheritance networks.

3.3. Knowledge Acquisition

The goal of the knowledge acquisition task is to obtain a formal representation of human knowledge that a computer can use at the symbol level in conformance with its meaning at the so-called knowledge level¹⁸. A same symbolic expression is given a computational interpretation at the symbol level like an instruction, and a semantic interpretation at the knowledge level by people like a piece of knowledge. That's why we can talk of programs by what they "mean" independently of the way they are programmed. In this framework, the CG formalism is exemplary insofar as it has been conceived to serve as an intermediate representation between natural language and logic. Although we can "read" a CG, this does not imply that a program does

the same. Thus, a clear distinction between the two interpretation modes must be kept.

In this context and as representing part of the knowledge, the same matters hold for taxonomies. A great attention must be paid to this important issue since types are the basic building blocks from which more complex representations are elaborated.

As a result, there are two related key issues pertaining to knowledge acquisition in the context of a CG application. The first one is the problem of acquisition of the type hierarchy. The second one is about the use of it. If the standard logical interpretation is clear, there are no means to control that the semantic, non formal, interpretation of a type hierarchy by people is the same. Every knowledge representation rely on the strong assumption that they are. When building the CTL of a domain, the designer has to tackle carefully this key issue (which becomes crucial with atomic types since only the genus are explicit but the differentiae are not).

4. Modelling an Ontology

When building knowledge based systems, the most important point is to ensure that the formal exploitation of the knowledge representation conforms to its meaning in the domain. The issues are twofold. On the one hand, the issue of knowledge representation is to design a formal representational system to account for the cognitive aspects of knowledge.

On the other hand, the issue of knowledge acquisition is to constrain knowledge so that it can fit into the formal representational system. To achieve this goal, a *normalization* of knowledge is a necessary step in which a theory of the domain, or *ontology*, must be elicited. Our claim is that, in order to remain consistent with the very nature of knowledge, the ontology of a domain can be represented by nothing but a taxonomic tree, precluding tangled hierarchies.

However, the formal structure to account for the ontology in the CG formalism is a lattice (section 3). To which extent, this lattice structure can be used to represent the ontology is therefore an important question. In order to answer this question, the knowledge representation fundamental issues are addressed below.

4.1. *Fundamental Issues of Knowledge Representation*

Generally speaking, AI consists in solving problems using knowledge that is usually expressed in natural language. Since there is no other operational expression of this knowledge, *i.e.* the domain is not formal, we have to tackle the semantic richness of language and its multiple potential interpretations. The basic problem is that the human knowledge we want to provide a program with is not formal. For example, there is no formal theory for modelling common sense knowledge or complex empirical medical knowledge. In medicine, though we can think, talk and reason about *pain*, there is no complete specification of this notion. The meaning of the notions we use

cannot be specified a priori, but only a posteriori, as we use them through experience. Moreover, in medicine, which is a non formal domain, many of medical notions do not even have the same definitions in different medical reference dictionaries^{36,37}. §

Knowledge people have about objects in a domain is mainly *descriptive*. It relies on the prototype notion whose logical status is not clear enough to be usable by a computer. People reason with this descriptive knowledge using the semantic laws of meaning such as analogy, metaphor, prototypical inference, etc. In non formal domain, we have no complete definitions for the categories of objects accounted by descriptive knowledge.

In order to have a computer deal with this kind of knowledge, we need to normalize it: we need to agree on the notions it mobilizes and on their use. Knowledge must be considered as an objective notion, *i.e.* shared. Basically, we have to fix the meaning of terms so that we all understand the same thing when we use a term. And so that we all agree when a computer infers on it. Knowledge will be adequately described if we account for explicitly rendering its objective and shared aspects. Normalizing is nothing but this. The use of terms will as a consequence no more rely on the full power of semantic interpretation, but on the manipulation of their explicit descriptions. Nevertheless, the constraints enforced on knowledge by the normalization process should not modify the meaning of the represented notions.

4.2. Normalizing by Necessary Conditions

An usual way of normalizing descriptive knowledge consists in stating the necessary relations between domain notions. Let us recall that type are categories of thought. Now, let us consider the notion of pain, which we all know and for which we want to confront our understanding of it. However, we have no mean to access all pains in the real world in order to see objectively whether we are able to agree or not on what pain actually is. The basic reason is that such a notion is *intensional*. Objects are considered through the way we think of them, not through their very nature, since this nature is not accessible, except for God.

The distinction between intension and *extension* was pointed out long ago by philosophers. The denotation of a type is its extension, *i.e.* the set of objects characterized by the type. Basically, there is no explicit relation between the intension of a type, its meaning, and its extension. Its extension is not a characterization of the type. Moreover, there are no means to yield the extension from intension in non formal domains.[¶] Citing Sowa³: “The type lattice represents categories of thought and the lattice of sets and subsets represents collection of existing things. The two lattices are not isomorphic and the denotation operator that maps one into the other is neither one-to-one nor onto.”

§ This is true at least in French. . .

¶ This is not the case in artificial domains.

The famous Frege's example of *Morning star* and *Evening star* illustrates this. The semantics, the intension of the concept *Evening star* is that, if I train my telescope on the first twinkling star in the evening sky, I shall see the so-called *Evening star*. Reciprocally, the last twinkling star in the morning sky is the so-called *Morning star*. It turns out that these two stars are the same, and correspond to the planet Venus. Hence, these two concepts are extensionally equivalent. But, obviously they are not intensionally equivalent. Observing the sky in the evening does not imply that I shall see the same thing when observing the sky in the morning. The concept of *observing the sky in the evening* does not analytically content the concept of *observing the sky in the morning* and it is not possible to deduce that one sees the same object.

Because of the lack of an explicit relation between the intensional definition of a type and its extension, one must be careful at not confusing the properties that characterize extensions with those that concern intensions. There are a priori no definitions of types relying on universal necessary and sufficient conditions; it is not possible to relate biunivocally objects to concepts of them. Since only intensions are available, we cannot but agree only on intensions in the normalizing process. For instance, pain can be considered as a human affection. If we accept this (partial) definition for pain, then we can all use this notion to understand each other. It must be noted that this necessary condition is still intensional, since the necessary inference from pain to human affection is drawn only by considering the meaning of pain and not by studying every individual pains existing in the world. Such subsumption based inferences do not rely on the inclusion of extensions but on the meaning of the notions. A first step of the normalization process is then to agree on and fix the necessary intensional conditions of types.

4.3. Normalizing by Necessary and Sufficient Conditions

Subsumption by the use of necessary conditions is the basis of all taxonomic organizations (see 2.1). It is a first attempt at normalizing a domain. A basic postulate when a type A subsumes B, is that A and B are intensionally different though B shares the intension of A. The properties of the genus A are inherited by B. The properties that make B different from A are also intensional. They correspond to its differentiae and are necessary to B. The laws that rule intensional subsumption are compatible with the minimal logical interpretation of a taxonomic link in case of such a partial definition. Such taxonomies, based on necessary conditions, are used for property inheritance. If it is known that an individual belongs to a given type, then (some) properties of its subsumers apply to it. But if the individual type is unknown, then it is impossible to classify it, to determine its type.

Taxonomies built from necessary conditions cannot allow classification. Their basic properties does not enable the addition of a new type while being sure there are no ambiguities. The defining conditions are not *sufficient* conditions: the differentiae between B and A are only necessary conditions. There is no equivalence between the

type and its definition. The result is that it is not possible to use the taxonomy either to look for a type through its definition or add a new type.

For instance, *action* is a type defined as *some event performed by some living being intentionally*. A *medical procedure* can be a subtype of *action* with the differentiae *performed by a doctor*. Now, if I know that some radiography, which is a medical procedure, was performed, then I know that this is an action that has been performed intentionally by a doctor. But, If I know that some action was intentionally performed by a doctor, then I have nothing else to say. This is not sufficient to decide whether this is a medical procedure or not. And I would be right, if Dr. John Doe kissed the nurse (his wife!).

Partial definitions are not enough to be sure that we talk about the same notions. We need sufficient conditions in order to agree that a notion correspond to the same knowledge. Being aware of the type subsumption nature is fundamental when one tries to understand, reuse or extend a given taxonomy – especially with atomic types when differentiae are not explicit. The knowledge normalization must be carried on in order to assign to types complete definitions.

Since types are intrinsically intensional, there must be a trade off to determine which properties are to be considered as *definitional*. This issue is relevant to every type, be it atomic or defined. The aim of the process is to build a theory of the domain. Normalizing by complete definitions is then extracting the *essence* of the notions that are used, *i.e.* their basic meaning on which we agree and that we can use in the same unique way.

4.4. Characterizing the Essential Properties

In order to assign to notions complete definitions, we aim at capturing the essence of these notions, thus at determining their *essential* properties. Capturing the essence of objects typically falls within the province of *ontology*. The term “ontology” comes from the philosophy and designs “the study of being as such”. An ontology may be defined as the set of objects that exist in a domain. So, building an ontology is to decide which objects one retains as existing in the studied domain. The notion of object is intensional; this step corresponds to conceptualization of the domain. Thus, defining types by deciding what are their essential characteristics amounts to build the ontology of the domain.

However, as pointed out previously, when the domain is not formalized, there are no methods that give the essential properties. One has to choose carefully the properties to consider as essential. A complete definition holds while no counter-examples occur. If we consider that an object belongs to a type but that it violates its definition, then the definition must be revised. As a consequence, the meaning of types is the result of a negotiation process. In one case, a given property is essential and the other not, and in some other case, the contrary occurs.

Let us consider for example the notions of *drug*, *biochemical substance* and *thera-*

peutic function. A pharmacologist knows that a drug is a biochemical substance, but not that it is a therapeutic function. Furthermore, a clinician considers that a drug is a function, but not a substance. As a consequence, a pharmacologist would know that SECTRAL is a drug made of *acebutolol* whereas a clinician that it has the betablocking function. Nevertheless, knowing that SECTRAL is made of *acebutolol* does not entail that one knows its function is the betablocking one. Worlds of pharmacology and clinic are not the same. If the pharmaceutical laboratory changes the biochemical molecule of SECTRAL, without changing its name nor its function, it remains the *same* object for the clinician while it is a new one for the pharmacologist.

This example shows that one must adopt a point of view, an interpretation, to be able to define the essential properties. As a result, a careful distinction must be made between essential and *incidental*, *i.e.* non essential, properties¹¹. This distinction is important to clarify what must be considered as the basic meaning of a type: its essence. The essential characteristics of objects are those such that, if a defined object loses only one of them, it no longer exists as such. These properties must be true by intension as long as the object exists. These properties are definitional, in the sense that objects that bear them are recognized as members of the type in every possible world.

Essential properties are opposed to properties an object can have or not, while remaining yet the same object. When defining a type, one must choose which properties are essential and, consequently, participate in the definition. Other, incidental, properties must not be considered as part of the definition. Among these ones, the *contingent* properties are not necessary; they are not true for all individuals, *eg.* deleterious for biochemical substances, or not always true for an individual, *eg.* pregnancy for women. *Peculiar* properties are always true for the individuals of a unique type. Such properties are sufficient, but not definitional. Man can laugh, although this property does not constitute the very essence of humanity.

The non essential properties of the defined types are not part of the ontology. Knowledge associated with them must be declared separately in a knowledge base. The associated inference mechanisms are then specific, *eg.* default reasoning for contingent properties. For instance, in MENELAS, properties considered as peculiar to a type are represented in the body of a CG schema (production rule) associated with the type. Contingent properties are represented as part of encyclopedic knowledge models.

4.5. *Essence and Taxonomy*

When the knowledge normalizing process is completed, types are defined in terms of necessary and sufficient conditions expressed through essential, intensional, properties. By definition, an essence is unique, and therefore a type can have but a unique definition.

However, unicity of essence can be argued as follows. Let us assume that a

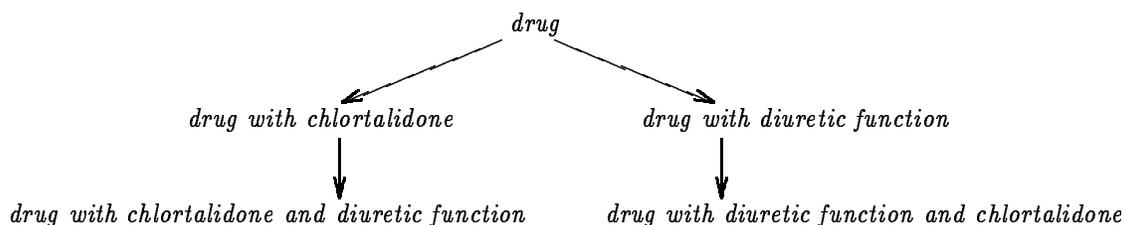
notion has two essences, then that means that there are two sets of necessary and sufficient conditions to determine it. We argue that either these two definitions are intensionally equivalent, or they determine two different notions. Namely, stating that the two intensions are distinct means there exists a possible world where the two essences denote different objects. The fact that we consider that both meanings are identical, rely on the assumption of a particular world where they denote the same objects. This property is contingent to this particular world but not implied by the definitions, then one of these definitions does not correspond to the essence of the notion.

Considering that we are characterizing ontological types by their essential properties in a non formal domain, we assume that a type has one and only one essence. Since the essence of a type consists in the set of its differentiating properties, this set is unique. In the resulting type subsumption structure, to each type corresponds a unique path from the root in the taxonomy. The immediate result is that the taxonomic structure is a tree and that a tangled type hierarchy cannot be build to represent an ontology.

A first consequence of the unicity of essence is that the extensional equivalence must not enforce intensional equivalence. The fact that extensional equivalence is realized in some world does not entail it is the case in every possible world. We could imagine one world where the two different definitions would correspond to different types. If we consider again Frege's example, the intension of *Evening Star* cannot tell that this concept is coreferential to the concept of *Morning Star*. In fact, one can think of a possible world in which these stars are different, while they coincide with the Planet Venus in our real world.

Considering that two intensions are the same amounts to forgetting knowledge we have about their actual extension in the world we consider. This knowledge is not part of the ontology. This is additional knowledge in the considered world that must be taken into account in a separate knowledge base.

Surprisingly, a second consequence is that the conjunction of intensional properties is not commutative along a path of the ontological taxonomy. The order is important in the definitions. Changing it, may change the meaning of the type. The main reason for this is that the intension of a given differentiating property, though expressed in the same terms, is not independent of its subsumers, which create an interpretation context of the term. Let us consider a medical example taken in our real world about the type *drug* and some of its subsumees.



In this example, the type *drug with chlortalidone* is defined with *drug* as its genus and the property P_c of *containing chlortalidone*, which is a *biochemical molecule*, as its differentiae. Type *drug with diuretic function* is defined from *drug* by the property P_f of *having a diuretic function*, which is a *therapeutic function*. On those two subtypes of *drug*, the differentiae P_f and P_c are respectively applied yielding the two leaf types of the figure.

At a first glance, these two types are intensionally equivalent; differentiating *drug* by P_c then P_f is the same as restricting *drug* by P_f , then P_c . Considering this, one would expect to join the two types into one, then forming a local lattice and considering that the two essences are simply redundant. For instance, HYGROTON is a drug that *contains chlortalidone*, and that *has a diuretic function*. Of course, it satisfies the two concurrent definitions. However, another drug called LOGROTON is also a *drug with chlortalidone and diuretic function*, but it is not a *drug with diuretic function*: in medical nomenclatures, LOGROTON is classified as *having a betablocking function*.

This real world counter-example illustrates that the two intensions, though close, are different. The reason why comes from the term *having diuretic function* which is not interpreted in the same way in the context of drug as a biochemical molecule, or in the context of drug as drug. In the former case, the *diuretic function* applies to the drug molecule taken in isolation, and de facto the *chlortalidone molecule* bear a diuretic function, and so does LOGROTON which contains some.

In the latter case, the *diuretic function* applies to a drug as a whole and as a therapeutic instrument. LOGROTON is not used to control the renal system, but the cardiovascular system. That is why it is not classified as *drug with diuretic function*, though it has some *diuretic* secondary effects due to the *chlortalidone* it contains. Hence, the term *diuretic function* is polysemic. The only way to avoid this mistake is to distinguish two different terms and state two different essences of functions: one for *biochemical molecules* and one for *drugs*.

Though, the temptation to merge two intentions on the basis of their same label notwithstanding the order in their expression must be prohibited. Many of such problems are due to the interpretative properties of natural language which is a strong bias in expressing intension but that we cannot avoid. The similarity between two definitions is only surfacic. We must assume that they are intensionally different, and then, correspond to two essences. The meaning of a property must be understood

through its position in the ontology; the same property in different positions, *i.e.* definitions, do not have subtly the same meaning.

In the following section, we propose some interpretation principles we have designed as a result of this theoretical reflection and through our experience in MENELAS at building the support of our CG based representations. These principles state how a taxonomy that aims to be an ontology must be built and read.

5. Ontology Acquisition Methodological Principles

It was shown in the previous section that the knowledge normalization process necessary to knowledge representation consists in building the ontology of the domain. In this ontology, which reflects the essences of the considered objects, a commitment is made about the intensional definitions of types. We also showed that the induced taxonomic structure is a tree. We present here methodological principles to control the acquisition of the ontology. These principles are constraints for building the ontology, and guidelines to read and use it. We now examine the principles that we are currently tentatively applying in the construction of the MENELAS CTL.

5.1. Principles of Structural Organization

A focal point for examining the issues of building taxonomies is the local system composed of a type and its subsumees, the *father* and its *children*. We shall call such a system a *local tree*. By stating principles applicable to every local tree, we can act on the whole structure. Each principle must be easily understandable and applicable by designers, so that it guides them in the ontology acquisition process. During this process, the intensional nature of types must be kept in mind.

Four principles are formulated as relations that must hold between types in a local tree. Each principle is accompanied by a description task: during knowledge acquisition, the designer that decides to give a type a new child must justify this decision by describing how this new child satisfies the principles. This justification is provided in natural language. Doing this exercise forces the designer to check that each principle is verified, by verbalizing the relationships that hold between this new type and its “neighbors” in the local tree.

The first two principles are directly related to Aristotle’s definition principles by genus and differentiae mentioned in section 3.2. They deal with the relations between a father and a child. The two others deals with the relations between siblings. The principles are exemplified by the addition of the type *mental state* as a subtype of *state*.

5.2. Similarity Principle

This first “similarity” principle controls the conformance of a type to its genus.

P1: Similarity Principle. A child must share the type of its father. This is the basic meaning of the taxonomic link. At the level of the whole local tree, this also means that all children share a common meaning. Being of the father's type is a necessary condition for a child.

Description: During knowledge acquisition, each time a type is given a new child, the similarities between the child and its father must be explicated.

Example: A *state* consists in a description of the world; *eg.*, it may be true or false. A *state* asserts something about objects in the world. A *mental state* consists in a description about *mental objects* which are objects of the world. As it shares a common meaning with *state*, type *mental state* is a subtype of *state*. Being a *state* is a necessary condition to be a *mental state*.

The kind of object involved in the *state* is therefore a good candidate for providing the semantic axis along which children of *state* will be defined. Type *mental state* specializes *state* along this axis.

5.3. Specificity Principle

The role of the second "specificity" principle is to justify the differentiae of a type with respect to its genus.

P2: Father-Child Difference Principle. A child must have a specific difference that distinguishes it from its father. This specifies a direction for the IS-A link. This specific difference in the context of the father expresses necessary and sufficient conditions for the child.

Description: The specific difference of the child. This often takes the form of a distinctive property that the child bears, or (*eg.*, for actions) as a specific thematic role always associated to the child.

Example: The specific difference of *mental state* relatively to *state* is that it involves *mental objects*, which are specific objects of the world. Being a *state that involves mental objects* is the complete definition of a *mental state*.

At this point, subsumption is taken into account and, since types are defined by necessary and sufficient conditions, the taxonomy can be used for classification.

5.4. Opposition Principle

The third principle of "opposition" further characterizes the relations between siblings. This principle is fundamental since the tree structure derives from it.

P3: Sibling Difference Principle. Siblings are organized in a system of oppositions. Each child of a type is opposed to the other children of the same type. All children must be pairly incompatible.

Description: The difference between a child and each of its siblings.

Example: Keeping the same example again and assuming that, *eg.*, a *physical state* is also a subtype of *state*, the difference between *mental state* and *physical state*, is that the former involves a *mental object* whereas the latter involves a *physical object*, both kinds of objects being distinct and incompatible in the ontology.

5.5. Unique Semantic Axis Principle

The last “unique semantic axis” principle provides a criterion on which children are to be compared to their father.

P4: Unique Semantic Axis Principle. In order to satisfy the two Difference Principles together (P2, P3), one can simply constrain all subtypes of a given concept type to differ from their father on a common dimension or axis, each subtype having an exclusive value for this dimension.

Description: The chosen semantic axis.

Example: For instance in the previous example, the decomposition of *state* is performed according to the kind of *objects* involved. There are three kinds of objects in the ontology: *ideal*, *physical*, and *mental* objects. *State* is then decomposed in three subtypes: *ideal state*, *physical state*, and *mental state*.

5.6. Using the Concept Type Tree

The combined use of the previous methodological principles at the node level in the taxonomy implies a local tree. The result is that they ensure that the whole taxonomy is a tree. The initial arbitrary “Concept Type Lattice” of conceptual graphs becomes the “Concept Type Tree” (CTT).^{||} Assuming that the type intensional definitions are really complete, this CTT becomes a decision tree. It can be used and understood.

Placing or searching a type in this ontological taxonomy can be done easily. The PlaceOrSearchType recursive protocol, described above in pseudo-code with its main steps, takes into account the four principles. Type is the type to be added or searched and Root is the father of the local tree. This protocol is initialized with the type to add and the top of the taxonomy.

^{||}This tree remains a lattice in the mathematical sense: a bottom node is kept for the absurd type.

PlaceOrSearchType(Type,Root)

0. **If** Type and Root are identical
 - Then** 1. Return(Type)
 - Else** 2. Control that Type shares similar properties with Root [P1]
 - 3. **If** a semantic axis has not been already determined [P4]
 - Then** 3.a. Determine a relevant semantic axis
 - 4. **Let** Dif be the projection of Type along this axis
 - 5. **Let** Sub be defined by Root then Dif [P2]
 - 6. **If** Sub is not a subtype of Root [P3]
 - Then** 6.a. Add a new subtype Sub to Root
 - 7. PlaceOrSearchType(Type,Sub)

Each step and principle must be controlled by the designer. We have developed KMT, a simple tool with a window interface, which goal is to help in controlling the acquisition of an ontology in the support of conceptual graphs according to the methodological principles we presented in this paper. This interface, written in Smalltalk, follows the guidelines of the methodology in order to manage, with the designer, the construction of the domain ontology. The results of the application of the four principles must be documented for each type. Pieces of knowledge related to types are captured here, then stored in CG knowledge bases. Afterwards, these knowledge bases are loaded in a CG processor³⁸ we developed for some modules of MENELAS.

MENELAS knowledge bases are currently under development. The ontology acquisition task is not yet finished. The domain is related to coronary diseases. Basically, we mainly aim at representing *medical acts* and *pathological states* of a patient with many related notions. In the current state, the “concept type tree” contains approximately 1300 types and the “relation tree” about 250 relations.

6. Discussion

6.1. More about the Four Principles

The two first methodological principles of similarity (P1) and specificity (P2) are nothing more than Aristotle’s principle for defining categories by genus and differentiae as presented in section 3.2.1. As it has been argued in 4.3, P1 enforces only necessary conditions, while P2 enforces necessary and sufficient conditions. Conformance to them is the foundation for type subsumption, and the resulting taxonomy can be used for classification.

The third opposition principle (P3) is fundamental here since the tree structure is a consequence of its application. Since it forces mutual exclusion between the sibling subtypes of a father, it does not permit that a given type be subsumed through two distinct paths from the father. This principle of opposition is imported from structural linguistics^{39,40}, where the meaning of a term consists in the differences

it has with others. Roughly speaking, to express something, we choose a linguistic realization and, as a consequence, discard the others. The meaning of the selected term comes from the fact that we invest this term with something specific that other terms do not have. In this sense, language is a system of differences, and so is knowledge expressed by this medium. In a similar way, a formal system such as the type hierarchy constitutes a system of differences: a type acquires its meaning from its position with respect to other types.

The last and fourth unique semantic axis principle (P4) seems desirable, but less crucial than the others. However, it is complementary to the opposition principle (P3) and helpful in practice. P3 imposes that sibling types be opposed. But guaranteeing that sibling types are intensionally opposed is closed to the issue we already discussed in section 4.5 about intensional equivalence. Knowing that intensions are distinct does not mean that they are opposed. Thus, the only way to know if two notions are opposite is to project them along the same semantic axis. In this respect, P4 constitutes an efficient principle to enforce the satisfaction of P3. Moreover, the choice of a unique semantic axis for decomposing all the proper subtypes of a type prevents the use of the same differentiating criterion further down. It is a mean to avoid the production of redundant definitions and temptations to confuse types. To a certain extent, the satisfaction of P3 can be viewed as a side effect of the satisfaction of P4.

6.2. Ontology Acquisition as Part of Knowledge Acquisition

6.2.1. Principles as Guidelines to Ontology Design

The four methodological principles described above aim at controlling that the taxonomic support of a CG representation reflect the ontology of a domain. We believe all these principles are necessary to structure the ontology, but not sufficient to complete its acquisition. They do not provide instructions, or at least strong guidelines, for eliciting an ontology. These principles are generally more constraining, or selective, than constructive. For instance, the fourth principle of unique semantic axis (P4) for decomposing a type, though not essential to the taxonomy structure, constitutes a strong but highly structuring constraint. Neither we did find methods to characterize relevant semantic axes for specializing a type, nor we were able to determine the order in which they should be considered. More precise principles for deciding whether to support a given decomposition of a type into subtypes would however be of great help in the design of concept hierarchies. However, it must be noted that, once the fundamental part of the ontology is already built up, it becomes easy to use it and augment it.

6.2.2. A Sharable Taxonomy

As explained in section 3.3, we strive to obtain the best possible correspondence between the computational properties induced by the structuring of the knowledge

representation and the properties that human interpretation assigns to this knowledge. In the absence of precise guidelines, humans may load concepts and their structuring with more meaning than is formally defined to the system (cf. 2.2.2). In our context, comments associated to each of the four principles provides documentation for each type, which is useful in several respects. They correspond to the commitment made on the type meaning, that can be shared by others. Such documentation indeed helps the maintenance of the support, when the designer updates the type hierarchy and needs to understand why a given decision has been made. It is also necessary if several persons have to work on the support. More generally, the documentation should help to understand, and then share, the actual meaning of each concept type more precisely than the first level of understanding resulting from the interpretation of its label.

6.2.3. An Help to Knowledge Elicitation

The constitution of the ontology has some virtues in view of knowledge elicitation. Because complete definitions must be assigned to types, all of their properties have to be elicited in order to decide which of them are to be considered essential. Brachman *et al.*¹¹ recommends the same with CLASSIC. Gaines and Linster¹⁰ propose tools to elicit entities and their attributes along certain dimensions.

As we pointed out in section 4.4, the remaining non definitional properties result from knowledge we have about a type in the world of reference. As a result, this knowledge must be recorded out of the ontology and indexed by the type. The use of a taxonomy is then a central structuring piece in a knowledge-based system as it provides guidelines for the acquisition of other pieces of knowledge. In MENELAS, besides the type and relation trees which support the ontology, we have several CG-based formalisms to account for such non-ontological knowledge, *eg.* schemata, relations signature, encyclopedic knowledge models.

6.2.4. A Word about Reusability

As mentioned in section 2.2.2, our own experience of reusability with UMLS did not encourage us to put forward the reusability of ontologies. Claiming that an ontology is reusable entails that it can be used to express knowledge for other tasks than the one for which it was designed. Noting the previous discussion in section 4.4, we advocate that the essential properties that are characterized, result from a point of view, an interpretation, of the domain notions. The consequence is that the ontology is dependent on this point of view. In non formal domains, like medicine, this point of view is likely dependent on the application.

Within CYC⁵, the use of common-sense knowledge has to go through the constitution of more specific “microtheories”, *i.e.* a set of sentences which are true in a certain context. So, the CYC microtheories constitute so many points of view about a domain which can be mutually inconsistent. This highlights the dependency between

an ontology and the context of its design.

If Gruber^{9,41} provides examples of KIF “reusable” ontologies, they are developed in formalized, mathematical domains. But, in such domains for which there exist formal models, concepts are defined analytically and only hold by/through their definitions. Since, they are not subject to interpretation, such ontologies can be therefore considered as reusable.

In non formal domains, the fact that there could be multiple ontologies of a domain questions their reusability. Nevertheless, starting from an existing ontology is easier than building one from scratch. Our methodological principles are particularly useful when trying to reuse an existing ontology^{1,29} and adapt it into a new framework.

7. Conclusion

In this paper, we have presented methodological principles to structure and acquire the ontology of a non formal domain. These principles come firstly from our experience in MENELAS at building and using the type taxonomy of a conceptual graph based, knowledge-based system, and secondly, from theoretical considerations on the intensional nature of types in knowledge representation. We claim that, in non formal domains like medicine, knowledge is rather descriptive than formal and therefore needs to be normalized in order to be used by a program while keeping the same meaning. This normalization consists in finding an agreement on the meaning of domain notions and in assigning to such intensional types definitions in terms of necessary and sufficient conditions at the knowledge level, so that the resulting taxonomy can be read and used for classification. Elaborating such definitions amounts to characterize the essences of types, *i.e.* to elicit the ontology of the domain. Assuming the unicity of the essence of an intensional type, we argue that the ontology can be nothing but a subsumption tree: a lattice structure cannot account for intensional type definitions; the problem comes from the difficulty to deal with intensional equivalence. Moreover, the properties of such an ontology guarantee that the intensional subsumption is compatible with the logical interpretation of the formal subsumption within the conceptual graph formalism.

The four proposed methodological principles derive from these considerations. They constitute constraints that aim at controlling both the nature of the subsumption links and the subsumption structure in order to elicit the ontology. The two first principles control subsumption following Aristotle’s principles of genus and differentiae. The others, the opposition between sibling types to enforce the tree structure. A simple window interface, KMT, assists the designer in consulting and building the ontology in the context of CGs.

The characterization of essential, definitional, properties is not a deterministic process. As a consequence, one must adopt a point of view on the domain notions. In our opinion, this fact has a serious impact on the possibility to reuse directly existing

ontologies. However, the simultaneous characterization of incidental properties is a mean to reveal other non ontological knowledge, and thus, participate in the knowledge acquisition process. Unfortunately, the methodological principles presented here only constitute guidelines since they are more selective than constructive: building an ontology still remains handicraft.

Finally, these results, though encountered within a conceptual graph application, are not specific to this choice. Since they rather deal with the ontology design issue, they also apply in the context of other knowledge representation languages based on taxonomies or definitions.

8. References

1. A. T. McCray, The UMLS semantic network, in SCAMC⁴² (1989), p. 503–507.
2. A. L. Rector, W. A. Nowlan, and S. Kay, Conceptual knowledge: the core of medical information systems, in Lun et al.⁴³ (1992), p. 1420–1426.
3. J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine* (Addison-Wesley, London, 1984).
4. R. J. Brachman and J. G. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive Science* 9:2 (1985) 171–216.
5. R. V. Guha and D. B. Lenat, Cyc: A mid-term report, *The Artificial Intelligence Magazine* (1990).
6. T. R. Gruber, The role of common ontology in achieving sharable, reusable knowledge base, in *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference*, eds. J. A. Allen, R. Fikes, and E. Sandewall (Morgan Kaufmann, Inc., Cambridge, 1991), p. 601–602.
7. B. Wielinga, W. van de Velde, G. Schreiber, and H. Akkermans, The CommonKADS framework for knowledge modelling, Technical report, Esprit Project KADS-II P5248, (1992). Excerpt of Deliverable 1.1.
8. J. Charlet, J.-P. Krivine, and C. Reynaud, Causal model-based knowledge acquisition tools: Discussion of experiments, in *Proceedings of the 6th European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, eds. T. Wetter, K.-D. Althoff, J. H. Boose, B. R. Gaines, M. Linster, and F. Schmalhofer (Springer-Verlag, Heidelberg, Germany, 1992), p. 318–336.
9. T. R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* 5 (1993) 199–220.
10. B. Gaines and M. Linster, Development of second generation knowledge acquisition system, in *Proceedings of the 4th European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, eds. B. Wielinga, J. H. Boose, B. Gaines, G. Schreiber, and M. van Someren (IOS Press, Amsterdam, The Netherlands, 1990), p. 143–160.
11. R. J. Brachman, D. L. McGuinness, P. F. Patel-Schneider, L. A. Resnik, and

- A. Borgida, Living with Classic: When and how to use a KL-ONE-like language, chapter 14 in Sowa⁴⁴ (1991), p. 401–456.
12. P. Zweigenbaum and Consortium MENELAS, MENELAS: an access system for medical records using natural language, *Computer Methods and Programs in Biomedicine* (1994). *To appear*.
 13. B. Nangle and M. T. Keane, Effective retrieval in hospital information systems, *Artificial Intelligence in Medicine* (1994). *To appear*.
 14. M. Cavazza, L. Doré, and P. Zweigenbaum, Model-based natural language understanding in medicine, in Lun et al.⁴³ (1992), p. 1356–1361.
 15. K. E. Campbell and M. A. Musen, Representation of clinical data using SNOMED III and conceptual graphs, in *Proceedings of the 16th Annual SCAMC* (Mc Graw Hill, Baltimore, 1992), p. 354–358.
 16. R. H. Baud, A. M. Rassinoux, and J. R. Scherrer, Natural language processing and semantical representation of medical texts, *Methods of Information in Medicine* **31** (1992) 117–125.
 17. M. Schröder, Knowledge-based processing of medical language: A language engineering approach, in *Proceedings of GWAI'92* (Bonn D, 1992).
 18. A. Newell, The knowledge level, *Artificial Intelligence* **18** (1982) 87–127.
 19. W. A. Woods, Understanding subsumption and taxonomy: A framework for progress, chapter 1 in Sowa⁴⁴ (1991), p. 45–94.
 20. F. Lehmann, Semantic network, in *Semantic Networks in Artificial Intelligence*, ed. F. Lehmann (Pergamon Press, 1992), p. 1–50.
 21. Department of Health and Human Services, Washington, *International Classification of Diseases — Clinical Modifications, 9th revision*, (1980).
 22. R. A. Côté, *Systematized Nomenclature of Medicine* (College of American Pathologists, Illinois, 1986). 2 volumes.
 23. B. L. Humphrey and D. A. B. Lindberg, Building the Unified Medical Language System, in SCAMC⁴² (1989), p. 475–480.
 24. A. L. Rector, A. Gangemi, E. Galeazzi, A. E. Glowinski, and A. Rossi-Mori, The GALEN CORE model schemata for anatomy: Towards a re-usable application-independent model of medical concepts, in *Proc MIE 94* (Lisbon, 1994).
 25. W. M. Tepfenhart, Using the situation data model to construct a conceptual basis, in *Proceedings of the 4th Annual Workshop on Conceptual Structures* (Detroit, 1989).
 26. M. Chein and M. Leclère, A cooperative program for the construction of a concept type lattice, in *Proceedings of the 2^{sd} International Conference on Conceptual Structures* (U. of Maryland, 1994).
 27. E. Aïmeur, METIS un système et une méthode d'explicitation de taxinomies destinés à l'identification de structure conceptuelle, PhD thesis, Université Paris 6, (1994).

28. M. R. Genesereth and R. E. Fikes, Knowledge Interchange Format version 3.0 reference manual, Technical Report Logic-92-1, Computer Science Department, Stanford University, (1992).
29. F. Volot, P. Zweigenbaum, B. Bachimont, M. Ben Saïd, J. Bouaud, M. Fieschi, and J.-F. Boisvieux, Structuration and acquisition of medical knowledge: Using UMLS in the Conceptual Graph formalism, in *Proceedings of the 17th Annual SCAMC* (Mc Graw Hill, Washington, 1993).
30. P. Zweigenbaum, B. Bachimont, J. Bouaud, J. Charlet, and J.-F. Boisvieux, Issues in the structuration and acquisition of an ontology for medical language understanding, in *Natural Language and Medical Concept Representation*, eds. C. Safran, C. Chute, and J.-R. Scherrer (Vevey, Switzerland, 1994).
31. J. F. Sowa, Towards the expressive power of natural language, chapter 5 in Sowa⁴⁴ (1991), p. 157–190.
32. M. Chein and M.-L. Mugnier, Conceptual graphs: fundamental notions, *Revue d'Intelligence Artificielle* 6:4 (1992) 365–406.
33. M.-L. Mugnier and M. Chein, Polynomial algorithms for projection and matching, in AWCG⁴⁵ (1992).
34. G. Ellis, Compiled hierarchical retrieval, in AWCG⁴⁶ (1991), p. 187–207.
35. R. Levinson and G. Ellis, Multi-level hierarchical retrieval, in AWCG⁴⁶ (1991), p. 67–81.
36. M. Garnier and V. Delamare, *Dictionnaire des Termes de Médecine* (Maloine, Paris, 1992).
37. Manuila, *Dictionnaire Français de Médecine et Biologie* (Masson, Paris, 1971).
38. J. Bouaud and P. Zweigenbaum, A reconstruction of conceptual graphs on top of a production system, in AWCG⁴⁵ (1992).
39. F. de Saussure, *Cours de Linguistique Générale* (Payot, Paris, 1985).
40. U. Eco, *Le signe; Histoire et analyse d'un concept* (Editions Labor, Bruxelles, 1988).
41. T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, (1993).
42. *Proceedings of the 13th Annual SCAMC* (IEEE, Washington, 1989).
43. Eds. K. C. Lun, P. Degoulet, T. Piemme, and O. Rienhoff, *Proceedings of MEDINFO 92* (North Holland, Amsterdam, 1992).
44. Ed. J. F. Sowa, *Principles of Semantic Networks* (Morgan Kaufmann Publishers, 1991).
45. *Proceedings of the 7th Annual Workshop on Conceptual Graphs* (Las Cruces (NM), 1992).
46. *Proceedings of the 6th Annual Workshop on Conceptual Graphs* (1991).