Knowledge Sharing and Reuse

H. Sofia Pinto

June, 2004
In this document we overview the Knowledge Sharing area. We present the problems that this area set itself up to solve and the solutions that were found. We then describe in some depth what is an ontology.
Chapter 1

Introduction

Since the early 1970’s it has been acknowledged that the key to build intelligent systems is the representation and storage of large quantities of knowledge. The knowledge base of an intelligent system is usually its most expensive module. Typically, for each intelligent system that was developed a new knowledge base was built from scratch. This was the reason why systems were kept small to medium size. As attempts were made to build larger and larger systems this cost was becoming prohibitive.

In order find solutions to cut down this cost, reusable components of knowledge based systems were sought. One of these components is the part of a knowledge base representing a common agreed upon vocabulary in one domain and the specification of its meaning. This component has been generically named an ontology. For instance, when one system for medical diagnosis and one for medical tutoring are built, the knowledge representing medical concepts is common to both systems. In this case, the knowledge in common to both systems pertains to the vocabulary in the medical domain. The existence of an ontology about the medical domain could avoid the need to re-represent the common knowledge for each system about that domain that is built. Another reusable component that was identified is the knowledge that pertains to solving problems. This component has been generically named a problem solving method. For instance, when one system for medical diagnosis and one for electronics diagnosis are built, a large part of the knowledge needed to solve diagnosis problems is common to both systems. This common knowledge needs to be re-represented for each diagnosis system that is built if libraries containing reusable diagnosis problem solving methods are not available.

In the early 1990’s a new area in Artificial Intelligence (AI) emerged, [29]. This new area, which is related to the Knowledge Representation area (KR) was generically named Knowledge Sharing. It aimed at [29]:
“Building knowledge-based systems today usually entails constructing new knowledge bases from scratch. It could instead be done by assembling reusable components. System developers would then only need to worry about creating the specialized knowledge and reasoners new to the specific task of their system. This new system would interoperate with existing systems, using them to perform some of its reasoning. In this way, declarative knowledge, problem-solving techniques and reasoning services would all be shared among systems. This approach would facilitate building bigger and better systems cheaply. The infrastructure to support such sharing and reuse would lead to greater ubiquity of these systems, potentially transforming the knowledge industry. . . .”

Four impediments to sharing and reuse of knowledge were identified [29]:

1. those arising from the existence of several knowledge representation paradigms and from the existence of a multitude of systems under the same paradigm,

2. the existence of profound differences among systems, even among those belonging to the same KR family or paradigm,

3. the lack of communication conventions that allowed run-time interactions between knowledge based systems and

4. the lack of shared vocabulary and domain knowledge.

To study and solve these problems the Knowledge-Sharing Effort [29] was proposed in 1990. This initiative was sponsored by the Defense Advanced Research Projects Agency (DARPA), the Corporation for National Research Initiatives and the National Science Foundation. The effort was organized into four groups each one responsible for providing solutions for each class of problem that was identified:

1. the Interlingua group,

2. the Knowledge Representation System Specification group,

3. the External Interfaces group and

4. the Shared, Reusable Knowledge Bases group.
The last group was also responsible for testing and analyzing the solutions provided by the four groups and coordinating the various groups.

The solutions found were [32, 40]:

1. translators of knowledge combined with an interlingua language,

2. knowledge representation system specifications,

3. agent communication languages and

4. ontologies.

In the following sections we present the historical context in which the area emerged, then we present each one of the challenges that the knowledge sharing area set itself to solve and discuss the solutions that were found and are currently under research to solve them.
Chapter 2

Historical context

Knowledge based systems, a short introduction to expert systems
See document “Sistemas Baseados em Conhecimento: Sistemas Periciais (breve resumo)”.
Chapter 3

Problems and solutions

3.1 Several knowledge representation systems

Since there are several knowledge representation systems and paradigms (different logics, frame systems, semantic networks, etc.), the knowledge represented in one particular system cannot be directly incorporated into another system. This problem is even more difficult to solve if those systems belong to different knowledge representation paradigms.

The obvious solution to the problem of having different knowledge representation systems under different knowledge representation paradigms is to develop translators between different knowledge representation systems. However, this has the problem that the number of translators required is $n(n - 1)$, where $n$ is the number of knowledge representation languages. Using an intermediate language as an interlingua to support translation can reduce the number of required translators to $2n$. Translation of knowledge between two knowledge representation systems is accomplished by translating knowledge represented in the source knowledge representation system language into the intermediate language and then from the intermediate language into the target knowledge representation system language. This intermediate language is the Knowledge Interchange Format (KIF) and was developed by the Interlingua group. One should note that KIF is not intended as a primary knowledge representation system language and it does not support any inference mechanism [9].

KIF had to have enough expressive power so that information between intermediate translations was not lost. It should be stressed that the translation process may require intermediate reformulation steps between recognition of the structure of the source sentence and generation of the target sentence [9]. Besides a structure that allows semi-automatic translation to
and from typical representation languages, the main functional requirements for this language were:

- that it had an agreed upon declarative semantics, independent of any given interpreter.

- that it was logically comprehensive, so that, one could write arbitrary sentences in first-order predicate calculus.

- that it provided mechanisms to represent metaknowledge (knowledge about knowledge).

- that it had sufficient expressive power to represent declarative knowledge contained in typical application system knowledge bases.

Although the language that was initially developed [12, 14] supported facilities to represent nonmonotonic, modal or higher order knowledge, in its last version, ANSI KIF [13],\(^1\) these features were removed. However, mechanisms were left to easily reintroduce such facilities in case of need.

The development of this language has stirred some waters. In [15] the limitations that the adoption of a language as a standard can impose on the development of new knowledge representation systems are discussed. Essentially the author feared that since KIF is similar to first order logic researchers would be constrained to develop similar systems to keep them as compatible as possible with other systems. He feared that those systems that could not change information via KIF with other systems would become isolated. The author defends that the language should be as flexible as possible to allow the easy introduction of new features.

Since the specification of ANSI KIF and its definition by the NCITS T2 committee on Information Interchange and Interpretation [28] the development of the language has ceased. However, this language is not dead. For instance, it is the underlying language of the most used ontology representation language, Ontolingua [16, 17] which is the underlying language of one of the ontology building tools with a large community of users, the Ontolingua Server [7].

A few translators between several knowledge representation languages and KIF were built, as reported in [40]. Translators from CLASSIC [5] and LOOM [26] into KIF and from Ontolingua (which can be viewed as an extension of KIF) out. As it was discussed in [40], building translators from a given knowledge representation language into KIF was easy because KIF

\(^1\)http://logic.stanford.edu/kif/
3.1. **SEVERAL KNOWLEDGE REPRESENTATION SYSTEMS**

has enough expressive power to represent declarative knowledge expressed in the tested source languages.

However, building translators from KIF to other knowledge representation languages is a considerable more difficult task [40]. One of the problems faced by the development of these translators is the fact that a given proposition may be represented in KIF in many logically equivalent but syntactically different ways and the fact that a single expression in the source language may be represented as several expressions in KIF. Therefore, a translator out of KIF may only be able to recognize and correctly translate a few of these forms. The other problem is the fact that, even if it is possible to translate out of KIF, the resulting representation may not be familiar or understandable to people familiar with that language. This happens because each language has certain conventions or styles of use that informally provide guidance as to how certain kinds of knowledge should be represented. These conventions are not a part of the language therefore, they cannot be captured in the KIF representation and get lost in the translation process.

One of the solutions to solve the two problems is to include human guidance in the translation process. Another solution is to extend KIF so that it supports additional constructs that are used in several knowledge representation languages (although these constructs have logical equivalents in KIF). Ontolingua was defined in this way. This language is as an extension of KIF to which the Frame ontology [16] was added. Therefore this language should preferably be used to translate into other knowledge representation languages belonging to the frame paradigm. However, one cannot say that the second solution is a complete success. The translator between Ontolingua and LOOM available at the Ontolingua Server was recently discussed [35] and the conclusion was that this specific translator was at draft level, that is, it requires a lot of human intervention after automatic translation. Other experiments between other languages point similar problems [34].

Only a few translators were developed, but due to the difficulty of the problem, they are still far from allowing full automatic translation between knowledge representation systems. Regarding KIF, the language is used by a few knowledge representation tools. However, it should be stressed that KIF was not primarily designed as an internal representation language within computers or in which humans should directly represent knowledge. Although KIF is used in several venues,² perhaps, it is not as widely used as initially planned, due to the difficulties of developing knowledge translators.

---

3.2 Differences within the same paradigm or family

Declarative knowledge representation systems can be grouped in different knowledge representation paradigms. There are several paradigms of knowledge representation systems: frame systems, semantic networks, etc. Within a given paradigm, there is a set of representation primitives that all systems have in common. For instance, the basis of all semantic networks is a set of nodes and a set of links. Nodes represent concepts and links represent relations. Nodes are connected among themselves by relations. Within each knowledge representation paradigm there are several families. For instance, within the semantic network paradigm there is the Conceptual Graph family [36] and the SNePS family. Different families implement the representation primitives that characterize a given paradigm differently. Moreover, the inference processes associated to those representation primitives are also different. A family is the result of the evolution of a given system due to research conducted by different researchers that were involved in the development of a given system.

Even between systems belonging to the same knowledge representation paradigm or even within the same family of knowledge representation systems there may be profound differences in syntax, semantics and aspects treated (for instance, one system may address nonmonotonic reasoning while another system from the same family may not).

The solution to the diversity among systems belonging to the same paradigm or even within a single family is the development of specifications for the representational component of knowledge representation families. These specifications define the representation language underlying one family and identify the features that every system belonging to that family should support. These features promote reusability of knowledge among systems belonging to the same family since they eliminate arbitrary differences among those systems. One should note that these specifications only include the minimal interface (one that is sufficient to create knowledge bases and query them in limited ways) and the minimal inference that those systems should perform. The goal was not to develop a system that had all the features within a given family but to find the basic primitives that should be supported by every system within a family and the inference mechanisms that should be supported for those basic primitives. The Knowledge Representation System Specification (KRSS) group developed such a specification [31] for the KL-ONE family (BACK [33], CLASSIC [5], KRIS [2], LOOM [26]).

The reasons why this family was chosen were:
3.3. LACK OF COMMUNICATION CONVENTIONS

1. the success that knowledge representation systems belonging to the description logic paradigm were and are currently having in the KR area,

2. the interest that this community already had for a common specification,

3. the large number of people involved in this Knowledge Sharing group that were already working with such systems and

4. the fact that the formal basis of those systems is particularly well suited to be used in this kind of specification effort.

As a result of the work carried out by this group, systems belonging to the description logics family can more easily be compared by stating which operators from the specification are implemented. Moreover, the common syntax incorporated in the specification allowed for easier comprehension of the literature in the area and the common semantics allowed some transfer of knowledge bases between systems [30].

Unfortunately, this specification work was not carried out for other families of knowledge representation systems.

3.3 Lack of communication conventions

The two previous problems involved reuse of represented knowledge. To use knowledge from one knowledge representation system there is no need to integrate it inside a second system if both systems can communicate. This prevents unnecessary duplication of information. The problem of duplication of information is transformed into a problem of communication between different systems.

To allow remote access a communication language is needed, that is, protocols for the exchange of represented knowledge among autonomous intelligent systems are needed. These high-level portable protocols play the same role to knowledge representation systems that SQL played to database management systems.

The communication language that was developed by the External Interfaces group is the Knowledge Query and Manipulation Language (KQML) [10] which allows not only communication between intelligent systems but also communication with conventional databases and object oriented databases. The key feature of this language is communication of attitudes about information, such as querying, stating, believing, requiring, achieving, subscribing
and offering. KQML is indifferent to the format of the information, thus KQML expressions often contain subexpressions in other knowledge representation languages.

In KQML messages are called performatives since the message is intended to perform some action by virtue of being sent. KQML assumes that [11]:

- agents are connected by unidirectional communication links that carry discrete messages,
- these links may have a non-zero message transport delay associated with them,
- when an agent receives a message it knows from which incoming link the message arrived,
- when an agent sends a message it may direct to which outgoing link the message goes,
- messages to a single destination arrive in the order they were sent,
- message delivery is reliable.

The semantic model underlying KQML performatives specifies that each agent appears on the outside as if it manages a knowledge base. Communication with the agent is made regarding its knowledge base, that is, questions about the content of the knowledge base, statements about what the knowledge base contains, requests to add or delete statements from the knowledge base, requests to use knowledge in the knowledge base to route messages to other appropriate agents. The implementation of the agent may not be necessarily structured as a knowledge base, for instance, it can be implemented as a simpler database scheme. Each agent manages its virtual knowledge base. Agents talk about the contents of their virtual knowledge bases and the virtual knowledge bases of other agents using KQML but encode statements in their virtual knowledge bases in whatever knowledge representation language they choose. KQML provides a standard set of interaction methods and protocols that are particularly suited to distributed programs. These can be viewed as knowledge based agents [11]. Tim Finin & Yannis Labrou [22] defined KQML semantics based on modal logic semantics. Some of the main arguments against KQML are the fact that it has been developed by a committee and its dependency on KIF.

There are several versions of KQML, such as the initial versions from around 1989, [10] and [23]. One of the main shortcomings pointed to KQML in its early days was the fact that its semantics was only informally and
3.4 Lack of a shared vocabulary and domain knowledge

Even if there weren’t any of the above listed problems the use of different representation primitives and terms to organize knowledge, and the use of different vocabulary to represent the same concepts or the use of the same vocabulary to represent different concepts, that is, the lack of a common vocabulary, would still be an unsurmountable problem.

Ontologies have been proposed to solve the problems that arise from using different terminology to refer to the same concept or using the same term to refer to different concepts. The term ontology has been borrowed from Philosophy. In Knowledge Sharing the meaning of this word is different from its meaning in Philosophy. Gruber [16] introduced the term ontology to mean an “explicit specification of a conceptualization” while in Philosophy Ontology means “a systematic account of Existence”.

3 The set of performatives is redundant.
4 http://www.fipa.org
5 “Ontology is the branch of Philosophy which deals with the nature and organization of reality. Aristotle defined it as the science of being as such.” [21]. The word Ontology was initially a synonym for Metaphysics. However, it has been divided into on the one
both meanings [21] proposed that Ontology should refer to the Philosophy meaning and ontology to the AI meaning.

An ontology specifies common vocabulary between different systems. It tries to identify and overcome the barriers to sharing and reuse of formally represented knowledge by AI programs that are due to a lack of consensus in what regards the vocabulary used and the different semantic interpretations in domain models. Informally an ontology consists of a set of terms and a set of constraints imposed on the way those terms can be combined. The latter set constrains the semantics of a term since it restricts the number of possible interpretations of the term. Terms in an ontology are a representation of concepts. For instance, the concept of a bachelor can be defined as a man that is not married. Here bachelor is the concept being defined and man and not married are the constraints imposed on it. We should stress that in an ontology concepts are represented, not words. Concepts, in general, are not specific of a given natural language [27].

Any knowledge based system has an ontology as one of its components, even if only implicitly. A knowledge base can only be assembled from smaller ones if their underlying ontologies are compatible and consistent one with the others.

The Sharing and Reusing Knowledge Bases group was responsible for:

- identifying the main research issues involved in knowledge sharing and reuse, which includes methodological problems, such as the development of methodologies to collaboratively build multidisciplinary knowledge bases, and engineering problems, such as scalability and shareability of those knowledge bases;

- developing ontologies that define terminology used to represent bodies of sharable knowledge which includes identifying what is important to be formally represented and make sharable, and defining coherent sets of terms that characterize the ontological commitments and the representation choices that were made while modeling those bodies of knowledge;

- coordinating collaborative knowledge sharing and reuse experiences between different research groups which includes not only trying to share and use knowledge bases built by others but also evaluate the use of ontologies as a sharing mechanism.

hand the study of the essence of beings and on the other hand the study and definition of a formal theory of the objects, that is, the study of the basic characteristics of all reality.
3.4. LACK OF A SHARED VOCABULARY AND DOMAIN KNOWLEDGE

To solve the problems arising from lack of vocabulary consensus, the Shared and Reusable Knowledge Bases group promoted the construction of several ontologies about the same and different domains. Their construction allowed not only a better understanding of the processes involved in developing ontologies, but also of the infrastructure and tools that are needed to develop, reuse and share ontologies. Criteria to guide the design of ontologies were found [18]. A tool specifically designed to build ontologies was developed, the Ontolinguia Server.\(^6\) The aim of this group included the development of libraries of sharable and reusable content knowledge. Examples of such ontologies are available in the Ontolinguia Server library, such as an ontology of time, an ontology about standard units [19] and an ontology of frames [16].

One can say that the solution to the problems arising from the use of different terminology involves the specification of methodologies to develop ontologies, the development of tools to build those ontologies and the development of ontologies.

\(^6\)http://www-KSL-SVC.stanford.edu:5915
Chapter 4

The ontology area

This area has been one of the most active ones within Knowledge Sharing. In the following section we discuss what ontologies are.

4.1 What is an ontology?

The term ontology has been used over the last years in AI with several meanings. In [21] one can find a discussion of some of the several meanings associated with the word in Philosophy, in AI, and particularly in the Knowledge Sharing area. The initial definition proposed by Gruber was slightly modified in [4] to “a formal specification of a shared conceptualization”. As discussed in [39] “conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group”. The broadest definition of ontology is “a vocabulary of terms and some specification of their meaning” [42, 41].

The main differences between both definitions lie on the formality requirement and on consensual nature of the knowledge represented in an ontology. It is important that the knowledge represented in the ontology has a consensual nature, at least among a given group, so that it can be reused in several knowledge based systems. After all, this is the main reason why ontologies are built. The formality requirement is not consensual in the ontology area. There are some ontologies which are expressed in a restricted and structured form of natural language that are nonetheless considered ontologies, for instance, the text version of an ontology about activities, processes,
organization, strategy [43]. There are even ontologies which are loosely expressed in natural language [41].

Therefore, ontologies form a continuum as proposed by Deborah L. McGuinness, Figure 4.1.

Ontologies are closely related to knowledge bases. The distinction between ontologies and knowledge bases lies on the different role played by represented knowledge. As we have seen, two reusable components of knowledge based systems were identified: ontologies and PSM. Both ontologies and PSM are the stable parts of a knowledge based system, that is, knowledge represented in these parts can be reused to build other knowledge bases. They can be used as a starting point to build other knowledge based systems. Ontologies are concerned with static domain knowledge, in contrast to problem solving methods which are concerned with dynamic reasoning knowledge. The knowledge base of a knowledge based system usually includes knowledge that typically changes with inferences. Knowledge represented in ontologies does not change with inference. Ontologies tend to represent knowledge that is more or less consensual of a community of people, whereas knowledge bases represent knowledge that is specific of the particular problem that the knowledge based system solves. Therefore, knowledge in ontologies is more appropriate to be reused and shared across applications. For instance, while an ontology on enterprise modeling contains concepts, such as activity, process, resource, in a knowledge base one would have represented the particular activities that are performed by a particular enterprise, the particular processes that take place in that enterprise, the actual process, activities, costs, resources that were used to build or produce a particular product, an estimate of the resources that were inferred to be needed to satisfy a new order that has just arrived.

Therefore, knowledge in ontologies is more appropriate to be reused and shared across applications.

Although ontologies aim at capturing static domain knowledge, it is generally acknowledged that an ontology depends on the application that powered its construction. If two applications are dealing with the same domain but the tasks they have to perform are different, then it is natural that the
ontologies they need about that domain are slightly different. Although most of the concepts are usually common they may be defined in different ways, such as with different levels of detail (as a class, a relation, etc.), capturing different points of view or features about the same concept (from a structural point of view, a functional point of view, etc.), with different levels of granularity. Different points of view may also imply that the same concepts are represented using different terminology.

It should be stressed that there is no single way of organizing concepts. There are different genuine alternatives. Therefore, one commits itself when an alternative is chosen.

Ontologies play an important part in communication between intelligent systems. Suppose that one application asks the other to perform a task. While transmitting information about the particular problem that it wants to see solved it must transmit that information in such a way that the other application can understand. Therefore, it may be important to translate information between different ontologies about the same domain.

Not only is compatibility among ontologies important in communication between different applications, but it is also important in building large systems from smaller ones. Any knowledge based system has an ontology as one of its components, even if only implicitly. A knowledge base can only be assembled from smaller ones if their underlying ontologies are compatible and consistent one with the others. If the ontologies underlying the different knowledge bases are not the same it means that either the domain is represented using different terminology or the same terminology is used with different meaning, that is, either the terms in each ontology are different or the axioms representing the constraints imposed on those terms are not equivalent. Only if the underlying ontologies of the knowledge bases are the same can they be assembled together. Therefore, one cannot build a large system by means of reuse if there is no understanding in what concerns the vocabulary that is used or if there is understanding about the vocabulary that is used but the axioms don’t represent the same statements or are in contradiction with one another.

To summarize, the solution to the problems arising from the use of different terminology involves (1) the specification of methodologies to develop ontologies, (2) the development of tools to build those ontologies and (3) the development of ontologies.

An ontology usually takes the form of an hierarchy of symbols. The symbols represent the concepts of a particular domain. Sometimes the hierarchy is referred to as a taxonomy and symbols are referred to as concepts, vocabulary or terms. However, this is not enough since these constituents could be interpreted differently by different systems. To restrict the possible in-
CHAPTER 4. THE ONTOLOGY AREA

ACTIVITY: something done over a particular TIME INTERVAL. The following may pertain to an ACTIVITY:

- has PRE-CONDITION(S);
- has EFFECT(S);
- is performed by one or more DOERS;
- is decomposed into more detailed SUB-ACTIVITIES
- entails use and/or consumption of RESOURCES
- has AUTHORITY requirements
- is associated with an [ACTIVITY] OWNER
- has a measured efficiency

DOER: The Role of an Actor in a Relationship with an ACTIVITY whereby the Actor performs (all or part of) the ACTIVITY.

Figure 4.2: Informal definitions in the ENTERPRISE ontology

interpretations of its symbols, an ontology includes a set of axioms. These axioms express the constraints that the symbols involved in those axioms must comply to. These axioms relate one symbol\(^1\) with the other symbols of the ontology. They restrict the possible interpretations for that symbol. Therefore, the most important part of an ontology is the semantics associated with its symbols, usually referred to as the content of the ontology.\(^2\) The content of an ontology is constrained through its set of axioms. Therefore, the basic unit of meaning is not a symbol but the theory, that is, the set of axioms that is associated with the several symbols in the hierarchy.

To give an idea of what an ontology looks like, we present the definitions of a few concepts from an existing ontology both in an informal and a formal way. In Figure 4.2 we show the text definition of an activity and a doer in the ENTERPRISE ontology [43]. An activity is characterized by the interval during which the activity takes place, its pre-conditions (what must be true for the activity to be performed), its effects (what is true once the activity is completed). There are also other attributes that characterize it, such as its doer, the sub-activities into which it can be decomposed. A doer is an actor that performs an activity. All concepts in upper case are also defined in the ontology. The definitions are expressed in a restricted and structured form of natural language.

In Figure 4.3 we present the definitions of those concepts implemented in

---

1 That is being constrained.

2 In opposition to form that usually is associated with its syntax.
4.1. WHAT IS AN ONTOLOGY?

(Define Frame Activity
:Slots
((Documentation "Something done over a particular Time=Range."
The following may pertain to an Activity;
* is performed by one or more Actual=Doer(s);
* is decomposed into more detailed Sub=Activity(s);
* Can=Use=Resource(s);
* An Actor may Hold=Authority to perform it;
* there may be an Activity=Owner;
* has a measured efficiency.")

((Instance Of Class)
(Subclass Of Activity=Der=Spec)
:TemplateSlots ((Actual=Activity=Interval (Minimum=Cardinality 0)
 (Cardinality 1)
 (Value=Type Time=Range))
 (Actual=Pre=Condition (Minimum=Cardinality 1)
 (Value=Type Pre=Condition))
 (Actual=Effect (Minimum=Cardinality 1)
 (Value=Type Effect))
 (Activity=Status (Minimum=Cardinality 1)
 (Value=Type Activity=State)))))

(Define Frame Actual=Doer
:Slots
((Documentation "The Actor in the Actually=Execute relationship.")
 (Instance Of Class)
 (Subclass Of Actor)
 :Axioms (\Rightarrow (Actual=Doer ?Actor)
 (And (Actor ?Actor)
 (Exists (?Activity)
 (Actually=Execute ?Actor ?Activity))))))

(Define Relation Actually=Execute (?Actor ?Activity)
 "A relationship between an Actor and an Activity
 Whereby the Actor has performed the Activity."
 :Def (And (Potential=Actor ?Actor) (Activity ?Activity)))

Figure 4.3: Formal definitions in the ENTERPRISE ontology
Ontolingua and kept in the Ontolingua Server library as Enterprise-Ontology. Activity and actual-doer (which corresponds to the concept of doer) were represented as classes.\(^3\) As it can be seen, there may be some differences between the implemented and the textual versions of an ontology. Activity is characterized by the interval of time during which it takes place, its pre-conditions, its effects and its status (done, to be done, etc.). Activity and actual-doer are related by the actually-execute relation. The actual-doer is an actor for which there is an activity to which that actor is related by the actually-execute relation. In this case the relationships used to establish the hierarchy of concepts are class-superclass and instance-class, for instance, actual-doer is a subclass of actor.

Ontologies can be classified, according to the issue of the conceptualization, usually referred as type, into [45, 20]:

- **representation ontologies or meta-ontologies**, capture the representation primitives used to formalize knowledge in a given knowledge representation family or system. For instance, the Frame ontology [16] which defines the terms that capture conventions used in object-centered knowledge representation systems (frames, description logics, etc.). This ontology defines concepts, such as class, relation, function, named-axiom, arity, exact-domain, exact-range, unary-relation, binary-relation. In this ontology, relations are sets of tuples (named by predicates), functions are a special case of relations, classes are unary relations (there is no special syntax for types), there is no special treatment of slots (since they can be represented as unary functions or binary relations) and classes are defined extensionally as sets (not descriptions);

- **general or upper-level ontologies**,\(^4\) classifies the different categories of entities existing in the world. Very general notions which are independent of a particular problem or domain are represented in these ontologies. Knowledge defined in this kind of ontologies is applicable across domains and includes vocabulary related to things, events, time, space, mereology.\(^5\)

An ontology of time, Simple-Time, can be found in the Ontolingua Server library. In Figure 4.4 we present part of its structure and in Figure 4.5 we present some of the definitions of the concepts represented in it. There are three main concepts: time point, time range and

---

\(^3\)Frames that are instances of Class.

\(^4\)They are also referred to as common/generic/top-level ontologies.

\(^5\)This ontology deals with the spatio-material properties of the physical objects, such as the relation part-of, proper overlap, interior part [3].
duration. A time point defines a single point in time that cannot be further decomposed. A time range is characterized by two time points, its starting and ending time points, and a duration. The ending time of a time range is equal to the sum of its starting time and its duration. A duration denotes a period of time and is characterized by a value and a measure. One can define an equality relation between two time points (if they represent the same time point) or two time ranges (if they have the same starting points and the same ending points). One can define several relations between time ranges, such as before, after, meets, overlaps.

Other examples of upper-level ontologies can be found in [37]. Some of the abstract upper-level ontologies presented in it were proposed by philosophers (for instance, Aristotle’s ontology) and others by knowledge engineers (for instance, Cyc’s [25] upper-level);

- domain ontologies, are more specific ontologies. Knowledge represented in this kind of ontologies is specific to a particular domain. These ontologies describe vocabulary related to a generic domain, such as airplanes, chemical elements, etc. They provide vocabularies about concepts in a domain and their relationships or about the theories governing the domain. For instance, the Plinius ontology [44, 38] is about the chemical composition of ceramic materials and Chemical-Elements [8] is an ontology about the chemical elements.

In Chemical-Elements, the most general concept, Elements, is characterized by its symbol, atomic number, chemical group, chemical period, atomic weight, boiling point, melting point, crystal structure, density at 20 degrees centigrade, electronegativity, etc. Ontologies are usually hierarchically organized. In Figure 4.6 we present part of the hi-
(Define-Class Time-Range (\?time-range)
"Time-range denotes a certain period of time. It consists of a start
and end time. A start time must proceed an end time.
Relations between Time-ranges are defined after James Allen's interval
relations."
;def (individual \?time-range)
:constraints (Equals (+ (Start-Time-of \?time-range)
(End-Time-of \?time-range)))

(Define-Function Start-Time-of (\?time-range) \?time-point
"(Start-Time-of \'tr) denotes a start time of a time range \'tr."
;def (and (Time-Range \?time-range)
(Time-Point \?time-point)))

(Define-Class Duration (?duration)
"Duration denotes a period of time. It consists of a value and a
measure"
;def (individual ?duration))

(Define-Relation Equals (?t1 ?t2)
"a time point ?t1 is equal to a time point ?t2.
a time range ?t1 is identical to a time range ?t2."
:axiom=def
((\=> (and (Time-Point \?t1) (Time-Point \?t2))
(\<=> (Equals ?t1 ?t2)
(\=> (\&\& (Year-of ?t1) (Year-of ?t2))
(\&\& (Month-of ?t1) (Month-of ?t2))
(\&\& (Day-of ?t1) (Day-of ?t2))
(\&\& (Hour-of ?t1) (Hour-of ?t2))
(\&\& (Minute-of ?t1) (Minute-of ?t2))
(\&\& (Second-of ?t1) (Second-of ?t2))))
(\=> (end (Time-Range ?t1) (Time-Range ?t2))
(\<=> (Equals ?t1 ?t2)
(\=> (Equals (Start-Time-of ?t1)
(Start-Time-of ?t2))
(Equals (End-Time-of ?t1)
(End-Time-of ?t2))))))

(Define-Relation After (?time-range=1 ?time-range=2)
"a time range ?time-range=1 succeeds a time range ?time-range=2."
;iiff=def (< (End-Time-of ?time-range=1)
(Start-Time-of ?time-range=2))
;equivalent (Before ?time-range=2 ?time-range=1))

Figure 4.5: Some definitions from the Simple-Time ontology
4.1. WHAT IS AN ONTOLOGY?

Figure 4.6: Part of Chemical-Elements hierarchy

...erarchy of Chemical-Elements. Dashed lines represent instance-class relations whereas solid lines represent class-superclass relations. Elements are divided into non-reactive and reactive. Helium, neon, argon, etc. are instances of non-reactive elements. They are all gases at normal pressure and temperature conditions. Reactive elements are further divided into metals, semi-metals and non-metals. There are several non-metal\(^6\) elements, such as carbon and oxygen. A few of them are grouped in the halogens subclass. The instances of semi-metal\(^7\) elements are boron, silicon, arsenic, tellurium, etc. Metals are further divided into transition metals and non-transition metals. Transition metals are divided into actinides, first, second and third transition series. While third transition series has several instances and one subclass, the lanthanides, all other classes are only related to their instances. Non-transition metals are divided into alkalines and alkaline-terrea which are directly related to their instances.

The term “domain” should be used to refer to the piece of reality that a particular domain ontology deals about. When we want to refer to the piece of reality that an upper-level ontology or a domain ontology deals about we should prefer the term subject [1]. However, domain and subject are used as synonyms in the area.

\( ^6 \)They are characterized by having high electronegativity.

\( ^7 \)They are characterized by a high electrical resistivity (ionization potential between 7.89 and 9.81 eV and electronegativity between 1.8 and 2.1).
Exercises

- Explain the main difficulties and problems that are faced when trying to share and reuse knowledge.

- Explain which solutions were proposed to solve the problems that are faced when trying to share and reuse knowledge.

- Which are the main advantages of sharing and reusing knowledge?

- The Knowledge Sharing and Reuse area has identified a few problems to solve and a few solutions for them. The language KIF and translators to and from KIF have been proposed to solve which problem? Which were the main criticisms made to the development of KIF?

- The Knowledge Sharing and Reuse area has identified a few problems to solve and a few solutions for them. Ontologies were proposed to solve which problem? Which was the solution proposed to allow communication between intelligent systems?

- Explain why the expressive power of the interlingua language KIF is important in the knowledge translation process between two systems with different knowledge representation languages.

- Comment the following assertion “I know a KBS that is an expert in (1) medical diagnosis, (2) symbolic derivation and integration, (3) planning the production of a computer assembly line, (4) configuring the computers to be produced in that assembly line, (5) monitoring and analyzing the geopolitical situation in Asia, Africa and the Middle East and (6) advising the external political to be followed by the US government.

- The most consensual definition of an ontology is “a formal specification of a shared conceptualization”. Explain why an ontology should be shared. Explain why the formality requirement is important.

- Ontologies are classified according to three types. Which ones? In which class can the Central Dogma ontology built in your project be classified? Justify your answer.
Bibliography


in the Knowledge Engineering Review, vol 13, Special Issue on Putting Ontologies to Use (eds. Mike Ushold and Austin Tate).

