Semantical and Ontological Considerations in Telos: a Language for Knowledge Representation

Dimitris Plexousakis
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
M5S 1A4
E-mail: dp@cs.toronto.edu

February 1993

Appears in: *Computational Intelligence, 9,1, February 1993*

Abstract

This paper focuses on the semantics of `Telos`, a language for representing knowledge about Information Systems. `Telos` is intended to support the development of information systems, especially in the requirements modeling phase.

An object-oriented representational framework is supported by `Telos`. Its features include *aggregation, generalization* and *classification*, the treatment of attributes as first-class objects and the explicit representation of time. `Telos` also provides an assertion sublanguage for expressing deductive rules and integrity constraints.

A possible-worlds semantics is defined for `Telos` knowledge bases. This semantics is intended to capture the peculiarities involved in the interpretation of temporal expressions. The integration of time has also inspired the treatment of *existence* in `Telos`. An ontology of objects based on the property of existence is proposed. In the spirit of `KRYPTON`, `Telos` knowledge bases are specified functionally, in terms of the operations provided for querying and updating them. This knowledge-level analysis will allow us to specify exactly what a knowledge base can be `ASK`-ed or `TELL`-ed about the domain of discourse. Soundness, Consistency and Completeness results have also been proven to complete the specification of `Telos` knowledge bases. This formal account of the language provides a logical framework that can be used to verify the correctness of any proposed implementation of the system.

Keywords: knowledge representation, semantic networks, knowledge bases, possible-worlds semantics, temporal knowledge

Subject category: Knowledge Representation

1 Introduction

Any language facility intended to support knowledge intensive applications, such as the development of information systems, should possess the ability to represent all the relevant knowledge about information systems. `Telos`[1] (Mylopoulos *et al.* 1990) is a knowledge representation language suitable for representing knowledge about information systems. As such, `Telos` has been developed to respect both the requirements of a knowledge representation formalism and those of a language intended to support "real world" applications.

Requirements for a knowledge representation language include a formal notation, a deductive mechanism for drawing inferences from a body of statements represented by means of this formal notation (a knowledge base) and a well-defined formal semantics that the deductive mechanism must respect in drawing inferences. On the other hand, a knowledge representation language must be expressive enough to be suitable for "real world" modeling.

The above requirements have shaped the representational framework supported by `Telos`. `Telos` supports an object-oriented representational framework whose features include *aggregation, generalization* and *classification*, as well as the treatment of attributes as first-class objects, and the explicit representation of time and of assertions for expressing deductive rules and integrity constraints.

In this paper we will focus on the semantics of `Telos` knowledge bases. Several languages based on semantic network formalisms lack - despite their name - a formal semantics. We will define a formal semantics for `Telos` knowledge bases, intended to capture the peculiarities involved in representing knowledge about the real world, especially when time is inherent in this knowledge. We will also elaborate on the ontological assumptions required for a semantic interpretation of `Telos`.

In the spirit of `KRYPTON`, `Telos` knowledge bases will be characterized functionally, in terms of the operations provided by the language for querying and updating them. In other words, `Telos` knowledge bases will be treated as instances of an *abstract data type* interacting with the user through a small set of operations (Levesque 1984). This knowledge-level analysis will allow us to specify exactly what the knowledge base can be `ASK`-ed or `TELL`-ed about the domain of discourse and how the system's beliefs are revised when an update is issued to the knowledge base. By means of this knowledge level characterization , the system's behaviour - with respect to the knowledge it can be considered to possess - can be predicted.

Section 2 of the paper briefly presents the features of the language, while in section 3 an ontology of objects for `Telos` is defined, based on the property of *existence*. In section 4 a semantics for `Telos` knowledge bases, based on a possible-worlds model, is formally defined and a partial solution to the problem of treating existence and non-existence in `Telos` is proposed. In section 5, a specification of `Telos` knowledge bases at the knowledge level is given. Section 6 presents some Soundness and Completeness results on this semantic account. We need to show that any answer to a query issued to the knowledge base can in fact be derived from the knowledge base by means of the axioms and the rules of inference, and, conversely, anything that can be derived from the knowledge base by means of the axioms and

[1] From the Greek word $\tau\acute{\epsilon}\lambda o\varsigma$ which means *end*; the object aimed at in an effort; purpose.

the rules of inference, can be provided as an answer to a query issued to the knowledge base.[2] Finally, section 7 summarizes the contributions of the language and its semantic account, and suggests directions for further research. The main points of this semantic treatment of `Telos` are summarized in the appendix at the end of the paper.

2 A Brief Introduction to Telos

2.1 The Representational Framework of Telos

The *representational framework* of `Telos` is a generalization of graph-theoretic data structures used in semantic networks, semantic data models and structurally object-oriented representations. `Telos` knowledge bases are collections of *propositions*. A *proposition* is the single representational unit provided by `Telos`. Formally, a proposition is defined as a quadruple with components *from, label, to* and *when*. These denote the source, label, destination and duration of the proposition respectively, and are propositions themselves.

`Telos` propositions are divided into two disjoint categories, *individuals* and *attributes*. *Individuals* in `Telos` will represent entities, either concrete or abstract, and can be thought of as synonyms of the terms entities, objects or nodes that have been used in other formalisms. *Attributes* will represent relationships between entities or between relationships, and can be thought of as links or relationships in semantic nets or semantic data models. Individuals and attributes are treated uniformly and are the building blocks of the structured objects of which a `Telos` knowledge base consists.

Propositions, individuals or attributes, are organized through the structuring principles of *aggregation, classification/instantiation* and *generalization/specialization*.

Aggregation is used for the construction of *structured objects*, which consist of collections of attributes that have a common proposition as source.

According to the principle of *classification/instantiation* every proposition must be *an instance* of more generic propositions, called *classes*. Since classes are themselves propositions, they also must be instances of more generic classes, and thus, an infinite dimension of classes along which propositions can be classified is created. *Instantiation* allows the grouping of propositions under generic descriptions, playing thus a role analogous to that of a frame in frame-based systems.

Along the infinite dimension of classes, propositions can be classified into *tokens*, propositions that have no instances, *simple classes*, having only tokens as instances, *metaclasses*, having only simple classes as instances, *metametaclasses* and so on. On the final level of the dimension reside ω-*classes*, classes having instances along more than one levels of the dimension.

Tokens are primarily intended to represent concrete entities in the domain of discourse, while classes can represent abstract entities.[3] `Telos` offers a number of built-in classes, one

[2] In fact, a limited form of completeness has been proven for Telos. Full-fledged completeness results are not claimed since the language is inherently incomplete (e.g. due to the ability to represent incomplete temporal knowledge).

[3] An ontology of objects has been defined for `Telos` in (Plexousakis 1990), where abstract and fictional

at each level of the dimension: Token, a simple class having all tokens as instances, S-Class, a metaclass or m1-Class having all simple classes as instances, M1-Class, a metametaclass or m2-class having all m1-classes as instances, and so on. The extension of each of these classes is the set of all propositions in the previous level of the dimension. A number of ω-classes is also offered to facilitate the structuring of the knowledge base: Proposition, having all propositions as instances , Class, having all classes as instances , Individual, having all individuals as instances , Attribute, having all attributes as instances, IndividualClass, having all individual classes as instances , AttributeClass, having all attribute classes as instances , and OmegaClass, having all ω-classes as instances.[4] Instantiation also determines the kinds of attributes an object that is an instance of one or more classes can have and the properties it must satisfy.

In an orthogonal direction to instantiation/classification, *IsA* hierarchies provide a means through which classes can be *generalized/specialized* (Borgida *et al.* 1984). Only classes residing on the same level of the instantiation dimension can be related through *IsA*. More specialized classes inherit all non-token attributes of the more general classes in a strict rather than default manner.

In the spirit of a uniform representation, both *Instantiation* and *Specialization* relationships are treated as propositions in Telos, by introducing the classes InstanceOf and IsA, which are defined as :

InstanceOf := <Proposition, instanceOf, Class, Alltime> and

IsA := <Class, isA, Class, Alltime>

where, Alltime is a constant representing an infinite time interval that includes all others.

The following example definitions of objects (borrowed from (Koubarakis *et al.* 1989)) depict the use of these features in the interaction with a knowledge base. A TELL command is used to introduce a new object in the knowledge base or to attach attributes to an already defined one. The token Krypton and the classes conf-paper, paper and paper-class are introduced along the first three levels of the instantiation dimension in an imaginary requirements model of a scientific conference.

```
TELL CLASS paper-class
     IN M1-Class
     WITH
      attribute
        card: 0..100
        paper-dscr: Class
END
```

```
TELL CLASS conf-paper
     IN paper-class
     ISA paper
     WITH
      paper-dscr
      referee: Referee
      reply-address: Address
END
```

object as well as objects existent in the past or the future, are taken into account within the Telos framework. The following section of this paper also presents briefly the treatment of the property of *existence* in Telos.

[4] A proposition should not be allowed to be an instance of itself in order to avoid paradoxes like Russell's paradox. However, an exception should be raised for some of the built-in classes of Telos, as e.g. Class or Proposition, which are propositions themselves and therefore must be instances of other propositions as well.

```
TELL CLASS paper
     IN paper-class
     WITH
      card
        :40
      paper-dscr
        author: Author
        title: String
        subject: subject-area
        date: TimeInterval
END
```

```
TELL TOKEN Krypton (at 1986/10 ..*)
     IN conf-paper (at 1986/10 ..*)
     WITH
      author
       first-author: Brachman (at 1986/10 ..*)
                   : Fikes (at 1987/1 ..*)
                   : Levesque (during 1986/10)
       title: 'A Hybrid Repr'n System'
       date: 1986/12
       subject: kr
END
```

The `IN` clause specifies the list of classes of which a defined object is to be an instance, whereas the optional `ISA` clause defines specialization and inheritance. `WITH`, which is also optional, is used to attach attributes to objects.

2.2 Representation of Temporal Knowledge

In many applications, it is desirable to represent the progression of states of the domain of discourse over a period of time. A knowledge representation system supporting time, such as `Telos`, must have the ability to represent the progression of states of the domain and provide answers to queries concerning both the current and previous states of the domain. In other words, it must have the ability to represent both the history of the domain and the system's knowledge of this history. For instance, if the process of software development is being modeled, different system versions need to be represented in order to show the process that led to a specific design.

In `Telos` we distinguish between *history time* and *belief time* indicating the history of the domain and the system's knowledge of this history respectively. The terms *valid time* and *transaction time* have been used in the area of temporal and deductive relational databases, e.g in (Snodgrass 1987), for history and belief time respectively.

The formal framework adopted for time in `Telos` is based on Allen's framework (Allen 84). The primitive notion is that of an interval. All possible relations between two time intervals can be expressed in terms of the relation `meets`, which is taken as primitive. The seven relations which, together with their inverses, are used to characterize any possible relative position of two time intervals, have been slightly modified when adopted by `Telos`. The temporal predicates offered in `Telos` are the predicates `during, startsbefore, endsbefore, rightbefore, before, costarts, coends, overlaps, equals` and their inverses `over, startsafter, endsafter, rightafter, after, costarts, coends, overlapped-by`.

Also, temporal constants as e.g. conventional dates and times, semi-infinite intervals with dates or times at one enpoint, and the special intervals `Alltime`, an infinite time interval including all others, `Now`, that represents the current system time, and `Null`, a null time interval, have also been included in the language. In addition, disjunction is disallowed in order to facilitate temporal reasoning.

In the previous example, the definition of `Krypton` depicts the treatment of time in `Telos`. The `TELL` command defining `krypton` informs the knowledge base that `Krypton` came into being at the beginning of October 1986 and is a paper since then. `Brachman`, the first author, started writing the paper at the beginning of the month and `Fikes` joined him on the first day of January 1987. Finally, `Levesque` was one of the authors during October 1986. Note that the temporal components of a `TELL` command are optional. If any of them is missing, then `(at systime ..*)` is assumed to be in its place by default, where `systime` is the time the above command is executed.

When a command like the above is issued to the knowledge base, the system starts believing the new knowledge from the time it is entered in the knowledge base and on. The belief time intervals that are assigned to propositions in the knowledge base are assumed to be persistent, starting at the time the command was executed, and the system keeps believing this knowledge until it is told otherwise.

2.3 The Assertional Component

Apart from the structural or frame component, used to specify knowledge base structures, `Telos` includes an assertional part that is used to state integrity constraints on the knowledge base, deductive rules and queries on the state of the knowledge base. The assertion language is a first-order language with equality. The terms of the language are: variables, constants and the functions *from(t), label(t), to(t)* and *when(t)* when applied to terms. Also, explicitly enumerated sets that are produced by set valued functions are terms.

The atomic formulae of the language and their corresponding meanings are as follows:

`prop (p,x,y,z,t)` : `p` is a proposition with components `x, y, z` and `t`

`instanceOf (x, y, t1, t2)` : `x` is an instance of `y` for the time period `t1` and is believed by the system for the time period `t2`

`isA (x, y, t1, t2)` : `x` is a specialization of `y` for the time period `t1` and is believed by the system for the time period `t2`

`att (x, y, t1, t2)` : `y` is a value of the attribute `att` of `x` for the time period `t1` and is believed by the system for the time period `t2`

`instanceOf (x, y, t)` : `x` is an instance of `y` for the time period `t`

`isA (x, y, t)` : `x` is a specialization of `y` for the time period `t`

`att (x, y, t)` : `y` is a value of the attribute `att` of `x` for the time period `t`

Also, for any temporal terms `x` and `y` in the language and every temporal predicate θ, $x\ \theta\ y$ is an atomic formula with the obvious meaning. If `x` and `y` are terms, then `(x = y)` is an atomic formula with the obvious meaning. Finally the well-formed formulae (wffs) of the language are defined recursively from the atomic formulae in the usual manner.

Closed formulae of the assertion language, or *assertions*, are integrated into the `Telos` framework and are represented as propositions. These propositions are instances of the built-in class `Assertion`. Any assertion can be attached to a proposition, called its *subject*, via the attribute mechanism of `Telos`. Hence, integrity constraints and deductive rules will be attached as attributes to propositions in the knowledge base and as instances of the built-in classes

IC:=<`Proposition`, integrityConstraint, `Assertion, Alltime`> and

DR:=<`Proposition`, deductiveRule, `Assertion, Alltime`>

Integrity constraints are to be interpreted as formulae that are entailed by the knowledge base during the history time they specify and according to the beliefs of the system during their belief time. The following example illustrates the use of the attribute mechanism to attach an integrity constraint to a `Telos` object.

```
TELL CLASS conf-paper
     IN S-Class
     ISA paper
     WITH
      integrityConstraint
        :$(Forall p/conf-paper)(Forall x/Author)(Forall t1,t2/TimeInterval)
          (author(p,x,t1,t2) ==> not (x MemberOf p.referee))$ (at 1988..*)
END
```

This constraint asserts that from 1988 and on, authors of a conference paper must not be referees for it.

Additional structure can be introduced for particular categories of propositions through the mechanisms offered by `Telos`. In particular, the ability to define attribute classes and metaclasses is offered, allowing, for example, to introduce common attributes of classes in the form of an attribute metaclass. In this way semantically similar attributes can be grouped together. Attribute metaclasses can also be used to state constraints on attribute values. The ability to talk about assertions is also provided. An extensive treatment of the extensibility of `Telos` and the features offered can be found in (Mylopoulos *et al.* 1990).

2.4 Interaction with Telos Knowledge Bases

The operations constituting the mechanism for interaction with `Telos` knowledge bases are the operations `TELL, UNTELL, RETELL, RETRIEVE` and `ASK`. A `Telos` knowledge base - `KB` for short henceforth- can be updated through `TELL, UNTELL` and `RETELL` and can be queried through `ASK` and `RETRIEVE`. All these operations operate on sets of propositions (`KB`'s).

`TELL` can be used to enter a new object in the knowledge base or to append new attributes to already defined objects. Its arguments are a knowledge base `KB` and a definition of a `Telos` object. It returns a new knowledge base `KB`′ that contains the definition of the new object and all its associated relationships that have been defined by `TELL`.

UNTELL can be used to inform the knowledge base that a specific relationship of an object that has been explicitly defined (i.e. a relationship defined by a TELL command) does not hold for a certain period of time. UNTELL's arguments are a knowledge base KB and a (not-necessarily complete) definition of the object and/or its associated relationships that are to be UNTELL-ed. The result is a new knowledge base KB$'$ which will reflect the system's change of beliefs.

RETELL is used to inform the knowledge base that a specific relationship that has been explicitly created for a certain object does not hold for a certain time period and some other relationship holds in its place. This relationship can be an instantiation, specialization or attribute relationship. The new relationship introduced by RETELL must be of the same kind as the one that RETELL specifies as a non-valid one for the specific time interval.

Queries to the knowledge base can be issued through the operations ASK and RETRIEVE. Their syntax is very similar and they only differ in that ASK answers queries by taking the deductive rules into account as well, while RETRIEVE considers only the structural information in the knowledge base.

More on these operations will be said in the sections to follow, where a functional specification of Telos knowledge bases at the knowledge level will be given.

3 An Ontology for Telos

In Knowledge Representation formalisms, such as Telos, the need to speak of *existence* and *non-existence* is apparent. We want the formalism to be as general as possible and suitable for real world applications.

The integration of time with propositions - the building blocks out of which knowledge bases are constructed - makes it possible to offer a rich ontology which distinguishes different types of existence along the lines suggested by Hirst (1989) . Existence, the way it is treated for knowledge representation/natural language understanding purposes at least, largely depends on the presence and adequate treatment of temporal knowledge.

This section presents and justifies the Telos ontology, and establishes the ontological assumptions required for a semantic interpretation of Telos. These are based, in their largest part, on the work of Hirst (1989) and McCarthy (1977, 1985) . Emphasis will be given on whether and how different types of existence can be handled within the Telos framework.

3.1 Existence and Non-existence

Hirst's main point is that existence can be predicated. However this is not an ordinary kind of predication. All kinds of objects, even concepts, ideas or fictional objects can be accounted for and are classified into one of nine types of existence. All these objects form the domain of discourse and thus the set of values over which quantifiers in expressions like $\forall x(...)$ or $\exists x(...)$ range. This set includes not only objects to which physical existence can be attributed, but all objects that form a part of the so called "naive ontology".

The main problem of previous approaches (Russell-Quine, Parsons, free logics) is the equivocation of existence. That is, they speak about existence as if all objects existed in the same manner. The assumption that Hirst introduces to address this problem is that all terms will have a denotation and all objects will be existent in some mode of existence. Everything is attributed existence but not everything is attributed physical actuality. We will adopt this assumption in our approach for an ontology for `Telos` although we will not account for all types of existence Hirst defines.

Let us first look at the nine types of existence considered by Hirst:

- Concrete physical existence in the present real world
- Physical existence in a past world as, e.g., that of Alan Turing
- Abstract, necessary existence, such as that of mathematical objects
- Doubly abstract existence, such as that of imaginary numbers
- Existence outside a world, as, e.g, that of God
- Abstract, contingent existence, as of e.g. freedom, ambition, happiness
- Existence as a concept
- Existence in a future world or unactualized existence, as that of an expected salary increase or that of an expected baby
- Existence in fiction, such as that of Zeus or Sherlock Holmes

In our definition of an ontology for `Telos` the universe of discourse will be considered as changing through time as entities are added to it. No object will cease to exist once it has started to, although its mode of existence might change. Again, existence is not to be confused with physical existence.

We will treat separately each of the above categories of existence that are also part of the `Telos` ontology, keeping in mind what Woods (1975) claims to be one of the requirements of a semantic network: the ability to represent various types of intentional objects without commitment to their existence in the external world, their external distinctions or their completeness in covering all the objects which are presumed to exist.

3.2 The Telos Ontology

In the definition of an ontology for `Telos` we will distinguish between objects that exist ephemerally - they have a lifetime that is represented by a finite or a semi-infinite time interval - at the present actual world or in a past or future world, and objects that exist eternally, such as God or abstract concepts and notions. Hence, we will be talking about *ephemeral* and *eternal* objects.

Ephemeral existence encompasses the following types of existence:
- Physical existence in the present actual world or in a past or future world.

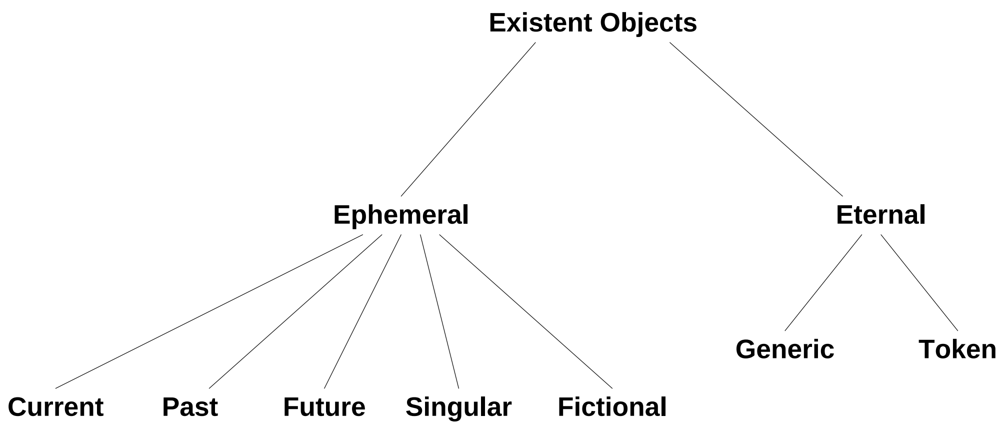

Figure 1: Classification of objects under the proposed ontology of Telos

- Abstract singular existence, the kind of existence Flight 501 to Athens or the movie "The Grapes of Wrath" have.
- Existence in fiction.

In the process of examining how to include fictional objects in the ontology of `Telos`, we were faced with the problem of deciding whether to classify fictional objects as eternal or as ephemeral objects whose lifetime is a right- infinite time interval. One could assume that a fictional object will be fictional at all time and in all possible worlds. This sort of reasoning applies well to abstract concepts and notions which are classified as eternal objects. In the case of fictional objects or book characters however, it is more realistic to assume that such objects exist at all times from the time of their initial conception and on. This assumption justifies the association of a right-infinite time interval with this category of objects.

The type of abstract singular existence is attributed to generic objects (classes) a particular instance or incarnation of whose can be considered as an ephemerally existing object and thus, have an associated finite time interval.

Eternal existence encompasses generic eternal and token eternal existence. The former is attributed to abstract concepts and notions, like freedom, happiness or existence. The latter characterizes objects existing outside a world, objects to which abstract necessary existence is attributed, and objects to which unactualized existence is attributed. Figure 1 summarizes the previous classification of objects in terms of their mode of existence.

In terms of possible world semantics, to each possible world a set of ephemerally existing objects will be assigned via a function $\mathtt{Q} : w_i \mapsto \mathtt{D}_i$, where w_i is a world corresponding to the time point i, $w_i \in \mathtt{W}$, the set of all possible worlds, and $\mathtt{D}_i$ is the set of existing objects in this world. The actual world is just one of the possible worlds. Let w_a denote the actual world. Then, the set $\mathtt{D}_a$ will be the set of real world objects. Hence, the universe of discourse $\mathtt{D}$ - and the range of quantification by consequence - will be equal to $(\bigcup_{w_i \in \mathtt{W}} \mathtt{Q}(w_i)) \cup \mathtt{P}$, where $\mathtt{Q}(w_i) = \mathtt{D}_i$, $\mathtt{P}$ is the set of abstract objects and $i \in \mathtt{T}$, a denumerable set of time points.[5]

[5] This is not a definition for the set $\mathtt{D}$. The definition of $\mathtt{D}$ is found in the next section.

Let's now see how each of the above kinds of existing objects are handled within the `Telos` framework.

The first three kinds, i.e. physical existence in the present, a past or a future world, are treated in the same fashion since time is a structural component of `Telos`. Objects that exist in the present actual world will constitute the set of existing objects $\mathtt{D}_a$ associated with the actual world w_a. Similarly, objects existing in a past or in a future world will constitute the set of existing objects associated with the possible worlds that correspond to the locations in time that specify a situation in the past or future. Hence, such objects can be defined as tokens in Telos with the associated time interval placing them in one or more possible worlds' existing objects sets. The time interval in the defining propositions of objects in Telos represents the lifetime of the object. Representation of such objects presents no additional burden in the Telos formalism and these objects can be used in quantified expressions just as much as objects of the actual world. Instances of the classes whose existence is characterized as abstract singular can also be defined as token in the language. Lastly, fictional objects are taken to be token objects in `Telos`.

As far as all other classes of objects are concerned, instances of generic eternal objects can also be defined as tokens in the language with the associated time interval being `Alltime`.[6] We'll follow McCarthy's (1985) approach where both abstract concepts, ideas and objects to which unactualized existence can be attributed, are treated as entities and are not formally distinguished from one another. Individual concepts such as the concept of a person are treated as classes (or individual classes) in `Telos`. Their instances however can be objects to which physical existence can be attributed.

A problem that arises now is how do we distinguish among all these kinds of objects and the different kinds of existence. An obvious solution is to predicate the different kinds of existence, thus introducing new built-in classes in the language. In the same manner we introduce *sorts* or *types* for all the categories of existence (one sort for fictional objects, another for abstract concepts and so on), thus transforming our logic into a many-sorted one. In this way typed constants, variables and predicates are allowed as well as sorted quantification. In addition *standard names* will be used for referring to the objects of the domain.

An ontology for `Telos` has been defined in this section in an attempt to achieve epistemological adequacy of some degree. According to McCarthy and Hayes (1969) "*practical systems require epistemologically adequate systems in which those facts which are actually ascertainable can be expressed*". Of course, the presented ontology is subject to controversy since the problem of defining an adequate ontology of objects has not received a universally acceptable solution. Our work was inspired by the presence of time in `Telos` which made possible the definition of an ontology based on the property of existence. Other ontologies could also be defined in orthogonal directions. For example, one could distinguish between physical and abstract objects, possible and impossible objects (Zalta 1983), tangible and intangible objects (Lenat and Guha 1989) or more general ontologies based on intrinsic properties of objects (Wand 1988).

[6] We are referring to history or valid time here. Belief or transaction time will denote when the system starts believing the existence of these objects.

The following section focuses in the construction of a possible-worlds model that will reflect the ontology defined in this chapter. This model will be used for expressing the semantics of Telos knowledge bases in the sections to follow.

4 A Possible-Worlds Semantics for Telos Knowledge Bases

A formal semantics for `Telos` knowledge bases is defined in this section. A knowledge base (`KB`) is considered as a partial specification of what the world is like, as opposed to the traditional view of the nature of a possible world as being a complete specification of what the world is, might be or might have been like. Hence, semantically, a knowledge base will be regarded as any set of possible worlds. In this model possible worlds will be indexed with respect to history time. That is to say that a possible world will represent an instantaneous snapshot of the domain of discourse over the course of time. This provides the ability to examine the state of affairs in the domain at any time. Belief time on the other hand, will be used to index knowledge-base slices or knowledge bases.

We begin the presentation of a semantics for `Telos` by building a possible- worlds model and establishing the assumptions that govern this semantic treatment of `Telos`.

4.1 A Possible-Worlds Model

The possible-worlds model to be defined is a variation of a standard *Kripke-structure* used mainly in modal logics of various kinds. A *possible-worlds model* M is defined as a 7-tuple

$\mathtt{M} = < \mathtt{W},\ \mathtt{D},\ \mathtt{Q},\ \mathtt{T},\ <,\ \delta, \mathtt{R} >$

W is a set of objects called "possible worlds". Possible worlds are entities in their own right. They can be considered as concrete particulars or, at least, made up of concrete particulars.[7]

D is the non-empty set of individuals constituting the domain of discourse. In the definition of an ontology for `Telos` we have distinguished among objects to which physical existence can be attributed and objects to which existence of some other mode (fictional, abstract etc.) can be attributed. The former have been called *ephemeral objects* while the latter have been called *eternal objects*. The set D - the domain of discourse - encompasses all objects to which one of the various kinds of existence the ontology accounts for can be attributed, i.e., the totality of objects `Telos` speaks of. For example both *Margaret Thatcher* and *Santa Claus* are members of D. The assumption here is that the elements of D are named by pairwise distinct *standard names* or *rigid designators* - as they are usually called in philosophical contexts, e.g. (Kripke 1972 ; Hintikka 1971) - so that no two standard names corefer. We also assume that an individual cannot split when we move from one possible world to another, i.e., the same standard name refers to the same individual in all possible worlds in which this individual is attributed some mode of existence.

$\mathtt{Q} : \mathtt{W} \rightarrow 2^{\mathtt{D}}$ is a function that assigns to each world $\mathtt{w}_i \in \mathtt{W}$ a certain subset $\mathtt{D_i}$ of D as its own peculiar domain. This set $\mathtt{D_i}$ is to be called the set of (ephemerally) *existing objects* at

[7] No commitment is made about the ontological status of the entities called "possible worlds". They are considered to be external to the defined ontology.

world $\mathtt{w}_i$, or the *universe* of $\mathtt{w}_i$. The sets of existing objects in the various possible worlds are partly identical (overlapping). Function Q thus makes possible to cope with the problem of *changing ontologies* (Reichgelt 1987) , i.e., the problem that the set of individuals over which one quantifies need not be the same from one moment in time (one possible world) to another (possible world). An additional requirement for function Q is to ensure that every individual - apart from the eternally existing ones - exists at at least one point in time. The set P of abstract objects can now be defined as $\mathtt{P} = \mathtt{D} - (\bigcup_{\mathtt{w}_i \in \mathtt{W}} \mathtt{D}_i)$, where $\mathtt{D}_i = \mathtt{Q}(\mathtt{w}_i)$.

T is a denumerable set of time points equipped with a binary order of precedence <, which is a linear order. The pair (T,<) is called a *temporal structure* or, as in (Goldblatt 1987), a *frame*. The intuitive meaning of < is that it represents an earlier-than relation and so it must be (at least) a partial order. A linear (non-branching), infinite and non-dense model of time is adopted. The model is non-branching, meaning that we examine the history of the domain of discourse over a single time-line. In addition, this time-line is isomorphic to the integer numbers (non-density). The following axioms express the properties of linearity, irreflexivity, non-density (discreteness) and succession respectively:[8]

`linearity` : $\forall x, y((x < y) \vee (y < x) \vee (x = y))$

`irreflexivity` : $\forall x(\neg(x < x))$

`discreteness` : $\forall x, y((x < y) \Rightarrow \exists z((x < z) \wedge \neg\exists w((x < w) \wedge (w < z))))$

`succession` : $\forall x \exists y(x < y)$

Clearly, the axiom of linearity excludes branching time with disjoint paths into the future. The fourth axiom can also be stated as:

$$\text{If } \ x < z, x < w, z < y \ \text{ and } \ w < y, \ \text{ then, either } \ z < w \ \text{ or } \ w < z$$

Together with the previous axioms it roughly says that the pair (x, y) can be thought of as the closed interval of points between x and y.

$\delta : \mathtt{I} \rightarrow 2^{\mathtt{W}}$ is a function associating possible worlds with time intervals. The set I is a set of time intervals defined as $\mathtt{I} = \{(t_1, t_2)/(t_1, t_2) \in \mathtt{T} \times \mathtt{T}, t_1 \leq t_2\} \cup \{\mathtt{Alltime}\} \cup \{(t, \infty)/t \in \mathtt{T}\} \cup \{(-\infty, t)/t \in \mathtt{T}\}$.[9] To every time interval i a subset of the set of all possible worlds will correspond. Since the view of a possible world as a snapshot of the domain of discourse at a specific time instance is supported, a whole set of possible worlds will correspond to a given time interval. By this definition of δ, $\delta(i)$, where $i = (t_1, t_2), t_1, t_2 \in \mathtt{T}$, will overlap with the sets of possible worlds corresponding to all time intervals overlapping with (t_1, t_2), contained in (t_1, t_2) or meeting (t_1, t_2). As was mentioned before, possible worlds are indexed with respect to time points and function δ provides such an indexing. In addition, function δ satisfies the following properties:

[8]The range of the quantifiers in these constraints is taken to be the union of the set **T** of time with the set $\{-\infty, \infty\}$, where $-\infty$ and ∞ are special time constants used to denote infinity.

[9]The $\leq$ relation can be defined from < and equality in the usual manner: $\forall x, y : x \leq y$ iff $(x < y) \vee (x = y)$.

- If $i_1, i_2 \in \mathtt{I}$ with $i_1 \sqsubseteq i_2$, then $\delta(i_1) \subseteq \delta(i_2)$[10]
- $\delta(Alltime) = \mathtt{W}$
- If $t \in \mathtt{T}$, $\delta(t, \infty) = \mathtt{W} - \bigcup_{t' \in \mathtt{T} - \{t'' | t'' < t\}} \delta(t, t')$

$\mathtt{R}$ is a binary reflexive relation well known in the literature of Modal Logic as the Accessibility Relation between possible worlds (Hughes and Cresswell 1972 ; Halpern and Moses 1985) . $\mathtt{w}_i \mathtt{R} \mathtt{w}_j$ means that the world $\mathtt{w}_j$ is accessible from the world $\mathtt{w}_i$. Intuitively, the accessibility relation specifies which worlds can be considered as alternatives of a particular one. In temporal logic contexts, as e.g. (Saake and Lipeck 1988) , $\mathtt{R}$ is also called a *reachability relation.* Usually, an accessibility relation is interpreted as : $\mathtt{w}_j$ *is accessible from* $\mathtt{w}_i$ *if* $\mathtt{w}_j$ *is conceivable from an intelligent agent situated in world* $\mathtt{w}_i$*.* In our case we define the accessibility relation between possible worlds as: $\mathtt{wRw}'$ *if and only if the time index of* $\mathtt{w}$ *temporally precedes the time index of* $\mathtt{w}$*'.* Hence, in our case, the accessibility relation $\mathtt{R}$ coincides with a $\leq$ relation on time indices and clearly is a reflexive, antisymmetric and transitive relation. It is reasonable and well-intuitive to assume that a possible world $\mathtt{w}$ is accessible by itself.

4.2 The Semantic Functions

The functions in terms of which the semantics for Telos will be expressed are introduced in this section. These functions will be denoted by $\mathtt{V}$,$\mathtt{E}$,$\mathtt{C}$,γ and σ.

$\mathtt{V}$ is an extension function mapping n-place predicate symbols to sets of ordered (n+1)-tuples of elements of $\mathtt{D}$ and $\mathtt{W}$. Function $\mathtt{V}$ will be used to check whether atomic formulae are satisfied in any given possible world. Atomic formulae will be allowed to take different values in different possible worlds. If the standard approach (semantics of first-order logic) was followed, and the extension of an n-place predicate ϕ simply was defined as a set of ordered n-tuples of elements of $\mathtt{D}$ (an n-ary relation over $\mathtt{D}$), then the value of every atomic formula would be the same in every possible world. Clearly this does not follow our intuition behind possible-worlds semantics. Hence, for every n-place predicate ϕ, $\mathtt{V}(\phi)$ can be expressed as

$\mathtt{V}(\phi) = \{< u_1, u_2, ..., u_n, \mathtt{w} > / u_i \in \mathtt{D}, i = 1, .., n \; and \; \mathtt{w} \in \mathtt{W}\}$.

$\mathtt{E}$ is an extension function mapping Telos classes to the corresponding relations over the domain of discourse $\mathtt{D}$ in the same manner as $\mathtt{V}$, that is, the last component of each tuple that is a member of this relation is a possible world belonging in the set $\mathtt{W}$ of possible worlds.

$\mathtt{C}$ is an extension function mapping defined classes to relations over $\mathtt{D}$ and $\mathtt{W}$, in the same manner as $\mathtt{E}$.

γ is a function mapping Telos propositions to their corresponding individuals in the domain of discourse. That is, if $p \in \mathtt{KB}$, then $\gamma(p) \in \mathtt{D}$.

Finally, σ is the function assigning meaning to propositions or sets of propositions (knowledge bases). We will use the notation $\sigma[\![\ldots]\!]$ to denote that σ is such a function.

[10] $i_1 \sqsubseteq i_2$ means that interval i_1 is contained in interval i_2.

4.3 Sorts in Telos

In this section we will introduce *sorts* and extend the language in which the semantics of `Telos` knowledge bases will be expressed, in order to account for the different kinds of existence we have talked about in the previous section. The expressions of this language, i.e., the expressions that are the result of applying function σ to `Telos` propositions, are in First Order Logic (FOL). The introduction of sorts transforms the language into a Many Sorted First Order Logic (MSFOL) (Enderton 1972) .

In a Many Sorted Logic constants, predicate and function symbols are typed and variables range over different universes. Each sort has its own universe, and the universes of the various sorts may or may not be disjoint. Since there is no equality relation between sorts, a possible non-disjointness can be regarded as accidental (Enderton 1972). However, we will allow for sorts to be non-disjoint under the assumption that an element of the language that can be assigned more than one sorts will be assigned the more specific one. The non-disjointness of sorts can also be justified by the non-disjointness of the corresponding classes. In addition, we will omit the sorts of constants, predicates, etc. in the expression of the semantics whenever it is clear which sort they are assigned.

As it will be clear later, when the semantics will be formally given, the incorporation of sorts, along with the introduction of new built-in classes for abstract, singular, fictional etc. objects in `Telos`, provides a solution to the problem of the distinction among the different kinds of objects and the various kinds of existence. For example, in order to take into account objects to which *fictional existence* can only be attributed, we will incorporate what we'll call "the sort of fictional objects". This provides a way of distinguishing objects as the above from other objects with similar representational properties, such as *abstract singular* objects.[11] The latter will be named by names of sort "Abstract Singular".

More formally, we assume a countably infinite and non-empty set `S`, whose elements are called *sorts*. This set contains:

- One element for every built-in class of `Telos` (such as `IndividualClass, Attribute, OmegaClass` etc.).
- One element for every class of objects to which one of the kinds of existence, other than physical, can be attributed. Such a class of objects is, for example, the class `Fictional`.

`S(C)` will denote the sort corresponding to class `C`. The built-in classes of `Telos` to which a sort in `SORTS` will correspond in each possible world, are the classes `Token, Attribute, IndividualClass, AttributeClass, Integer, String, TimeInterval, Token, S-Class, M1-Class, M2-Class ...OmegaClass`. The classes of objects the ontology of `Telos` is made up from and to which only non-physical existence can be attributed are the classes `Fictional, Abstract singular, Abstract necessary` and `Abstract contingent.`[12] Constants, vari-

[11] Recall that both fictional and abstract objects have right-infinite associated time interval.

[12] Abstract necessary existence is the kind of existence we consider mathematical objects to have, while abstract contingent includes existence of abstract concepts, existence as a concept, existence outside a world and unactualized existence.

ables, function and predicate symbols are assigned sorts by the usual definitions in (Enderton 1972). In addition terms are recursively assigned unique sorts.

4.4 Semantics

We start with the semantics of the initial `Telos` knowledge base, $\mathtt{KB}_0$. The initial knowledge base contains no domain-specific information, i.e, no information about what the world is like. It is reasonable thus to assume that the semantics of an empty `KB` contains all possible specifications of what the world might be like. Hence, we define: $\sigma[\![\mathtt{KB}_0]\!] = \mathtt{W}$, where `W` has been defined as the set of all possible worlds in the model `M`.

Initially however, the knowledge-base contains built-in information about the representational framework of `Telos`. The initial theory of a `Telos KB` contains an axiomatization of time based on Allen's proposal, as well as the basic framework axioms. The axiomatization of time specifies the primitive and derived relations of a time model based on time intervals. The basic framework axioms establish the relations between the built-in components of `Telos` and the relations `IsA` and `InstanceOf`. All of the above axioms must hold in the initial knowledge base as well as in all consecutive knowledge bases.

For an individual token proposition `p`, we define the semantics of `p` to be: $\sigma[\![\mathtt{p}]\!] = \gamma(p)$, and $\gamma(p) \in \mathtt{D}$. Moreover, if $p \in \mathtt{KB}$, then $\sigma[\![\mathtt{p}|\ when(\mathtt{p}) = i]\!] = \gamma(\mathtt{p})$ and $\forall w \in \sigma[\![\mathtt{KB}]\!]\ \gamma(\mathtt{p}) \in \mathtt{Q}(w)$ iff $w \in \delta(i)$.

The intuition behind this definition of a semantics for individual token propositions is that, first, their denotation must be an element of the domain of discourse and, second, this denotation must be in the set of existing objects in the corresponding possible worlds. However, we must treat differently individual token propositions for which $when(p) = \mathtt{Alltime}$. In this case the corresponding individuals belong to a category of objects to which existence other than ephemeral is attributed according to the ontology we have defined for `Telos`.

If `p` is an *attribute token* proposition, then the semantics of `p` will be a relation over the Cartesian product of the extensions (or denotations) of the source and destination components of `p` and the set `W` of possible worlds. Intuitively, this indicates when the specific relationship between the individuals that serve as source and destination of `p` or the members of the classes `from(p)` and `to(p)` holds. More formally, if `p` is an attribute token proposition and $\mathtt{p} \in \mathtt{KB}$, then $\gamma(\mathtt{p}) \in \mathtt{D}$ and $\sigma[\![\mathtt{p}|\ when(\mathtt{p}) = i]\!] \subseteq \mathtt{D} \times \mathtt{D} \times \mathtt{W}$. The following definition describes the tuples that constitute the extension of an attribute token proposition.

$\sigma[\![\mathtt{p}|when(\mathtt{p}) = i]\!] = \{< o_1, o_2, w > \mid o_1, o_2 \in \mathtt{Q}(w) \cup \mathtt{P}, o_1 = \gamma(from(\mathtt{p}))\,, o_2 = \gamma(to(\mathtt{p}))$ and $w \in \delta(i)\}$

Note that in the above definition, no restriction is imposed on the relation between the intervals associated with propositions `p` and `q` and the interval i of the attribute proposition. This allows for the definition of properties of objects, where the ontological status of the property is different than that of the objects denoted by `p` and `q`.

If `p` is a proposition in the `KB` defining a simple class (individual or attribute), $\gamma(p)$ will be required to be a member of `D`. Two reasons support this requirement: The first reason is purely syntactic. We want a class to participate in the extension of more generic classes and in tuples of the relations representing the extensions of the `IsA` and `InstanceOf` relationships.

The second reason is that, according to the already defined ontology of Telos, we want to represent individual concepts such as *person*, as first class individuals, and individual classes serve this purpose as well.

If p is an individual simple class, then the extension of p is E(p). If p is an attribute simple class, then the extension of p will be a relation consisting of tuples whose the first two components are individuals in the domain D and the third a possible world. To be more specific, the first component will be the standard name of the *source* component of p and the second the standard name of the *destination* component of p. The third component will be a possible world belonging to the set of possible worlds corresponding to the *duration* component of p. Hence:

$\sigma[\![\mathtt{p}|when(\mathtt{p}) = i]\!] = \{< o_1, o_2, w > |o_1, o_2 \in \mathtt{D},\ o_1 = \gamma(from(\mathtt{p})), o_2 = \gamma(to(\mathtt{p})),$
$< o_1, w > \in \mathtt{E}(from(\mathtt{p})), < o_2, w > \in \mathtt{E}(to(p))$ and $w \in \delta(i)\}$

According to this definition, for any possible world whose corresponding time instance falls within the time interval associated with an attribute simple class, there will be a tuple in the extension of the class having as first and second component the standard names of the class's source and destination components respectively, and as third component this possible world.

As was the case for simple classes, for any Telos proposition p defining a metaclass, we require that there exists an element d of the domain D, for which $\gamma(\mathtt{p}) = \mathtt{d}$. The same reasons as in the case of simple classes justify this requirement. For this element d, it is not necessarily the case that there exists a possible world $w \in \mathtt{W}$ such that $d \in \mathtt{Q}(w)$. We would rather say that it is unlikely to have a metaclass representing an ephemerally existing entity.

The semantics must also capture instantiation and specialization relationships among Telos objects. An instantiation relationship is represented in the extension C of classes. If q is an instance of p for a time interval j, then C(p) contains a tuple $< \gamma(\mathtt{q}), w >$ for all $w \in \delta(j)$. The extension of the built-in class InstanceOf, E(InstanceOf), contains a tuple for any proposition defining an instantiation relationship for all possible worlds in the associated time interval. Specialization is accounted for by defining the extension of IsA to be $\mathtt{E(IsA)} \subseteq \mathtt{D} \times \mathtt{W}$. If p and q are two classes in a Telos knowledge base and class q is a specialization of class p for a time interval i and r is the proposition defining this relationship, then $\forall w \in \delta(i)\ \ < \gamma(\mathtt{r}), w > \in \mathtt{E(IsA)}$.

In the semantics of Telos, since every proposition is represented by a standard name in the domain of discourse, we only need to account for belief time by indexing our knowledge bases with respect to it. To be more specific, we introduce a relation B over $\mathtt{D} \times \mathtt{I}$, where I has been previously defined as the set of all possible time intervals. The first component of B is a standard name for a Telos proposition already in the knowledge base. The second component is a time interval (usually a right-semi-infinite one) indicating for which time interval the specific proposition is believed by the system. This time interval specifies a knowledge base or a slice of a Telos knowledge base containing all propositions believed by the system for this time interval, and thus, provides a means of indexing knowledge bases with respect to time.[13] We now formally define the notion of a formula being satisfied for

[13] Here we treat *belief* as a relation. It could however be treated as a modality as in (Rao and Foo 1989) ,

a possible-worlds model $\mathtt{M}$ and a world $\mathtt{w}$, via the relation $\models$, a binary entailment relation between model-worlds pairs and formulae.[14]

$(\mathtt{M}, w) \models \mathtt{prop(p,x,y,z,t)}$ iff $\forall w \in \delta(t) < \gamma(p), \gamma(x), \gamma(y), \gamma(z), \gamma(t), w > \in \mathtt{V(prop)}$ and $\exists \mathtt{q} \in \mathtt{KB}$ such that $from(\mathtt{p}) = x$, $label(\mathtt{p}) = y$, $to(\mathtt{p}) = z$, $when(\mathtt{p}) = t$ and $\gamma(\mathtt{q}) = \gamma(\mathtt{p})$.

$(\mathtt{M}, w) \models \mathtt{instanceOf}(x, y, t1, t2)$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(instanceOf)}$ and $\forall w' \in \delta(t1) < \gamma(x), w' > \in \mathtt{C}(y)$ and $\exists \mathtt{p} \in \mathtt{KB}$ such that $from(\mathtt{p}) = \mathtt{q} =< x, instanceOf, y, t1 >, label(\mathtt{p}) = instanceOf$, $to(\mathtt{p}) = \mathtt{InstanceOf}, when(\mathtt{p}) = t2$ and $< \gamma(\mathtt{p}), w' > \in \mathtt{E(InstanceOf)}$, $< \gamma(\mathtt{p}), t2 > \in \mathtt{B}$.

$(\mathtt{M}, w) \models \mathtt{isA(x,y,t1,t2)}$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(isA)}$, and $\exists \mathtt{p} \in \mathtt{KB}$ such that $\mathtt{p} =< \mathtt{x, isA, y, t1} >$ and $\forall w \in \delta(t1)$ $, < \gamma(\mathtt{p}), w > \in \mathtt{E(IsA)}$ and $\exists \mathtt{q} \in \mathtt{KB}$ such that $from(\mathtt{q}) = \mathtt{r} =< \mathtt{x, instanceOf, IsA, t1} >, label(\mathtt{q}) = instanceOf$, $to(\mathtt{q}) = \mathtt{InstanceOf}), when(\mathtt{q}) = \mathtt{t2}$ and $\forall w \in \delta(t2)$ $< \gamma(\mathtt{r}), w > \in \mathtt{E(InstanceOf)}$ and $< \gamma(\mathtt{q}, t2 > \in \mathtt{B}$.

$(\mathtt{M}, w) \models \mathtt{att}(x, y, t1, t2)$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(att)}$ and $\exists \mathtt{q,r,s,Q} \in \mathtt{KB}$ such that $\mathtt{q} =< \mathtt{x, att, y, t1} >, \mathtt{r} =< \mathtt{q}, instanceOf, \mathtt{Q}, t3 >$, $\mathtt{s} =< \mathtt{r}, instanceOf, \mathtt{InstanceOf}, t2 >$, $\mathtt{Q} =< \mathtt{CL}, \ldots, \mathtt{Class}, \ldots >$ and $\forall w \in \delta(t3)$ $< \gamma(\mathtt{r}, w > \in \mathtt{C(Q)}$ and $\forall w \in \delta(t2) < \gamma(\mathtt{s}, w > \in \mathtt{E(InstanceOf)}$ and $< \gamma(\mathtt{r}), t2 > \in \mathtt{B}$ where $\mathtt{CL}$ is a class of which x is an instance for a time interval containing $t1$.

The satisfiability of the rest of the atomic formulae is defined similarly. Equality of terms and temporal predicates must be treated in a special manner:

$(\mathtt{M}, w) \models (t1 = t2)$ iff their corresponding individuals are both in $\mathtt{Q(w)}$ or both in $\mathtt{D}$ and they are named by the same standard name.

If x, y are time intervals with $x = (x_1, x_2)$, $y = (y_1, y_2)$, then $(\mathtt{M}, w) \models x \; \theta \; y$ iff $(\mathtt{M}, w) \models \tau(\theta, x, y)$, where $\tau(\theta, x, y)$ is a point-based expression.

Having defined the satisfiability of atomic formulae, we now proceed to defining satisfiability for the well-formed formulae of the assertion language. For the propositional subset of the assertion language the semantics of the well-formed formulae are defined as usual.

$(\mathtt{M}, w) \models \alpha \; \mathtt{and} \beta$ iff $(\mathtt{M}, w) \models \alpha$ and $(\mathtt{M}, w) \models \beta$

$(\mathtt{M}, w) \models \alpha \; \mathtt{or} \beta$ iff $(\mathtt{M}, w) \models \alpha$ or $(\mathtt{M}, w) \models \beta$

$(\mathtt{M}, w) \models \alpha \Rightarrow \beta$ iff $not((\mathtt{M}, w) \models \alpha)$ or $(\mathtt{M}, w) \models \beta$

$(\mathtt{M}, w) \models \alpha \Leftrightarrow \beta$ iff $(\mathtt{M}, w) \models \alpha \Rightarrow \beta$ and $(\mathtt{M}, w) \models \beta \Rightarrow \alpha$

$(\mathtt{M}, w) \models \mathtt{not} \; \alpha$ iff $not((\mathtt{M}, w) \models \alpha)$

where the *Coherent Theory of Belief Revision* is examined. The semantics of this theory is based on a modal logic of time-dependent beliefs and a modal operator *BEL* with arguments a propositions $\mathtt{p}$ and a time-instance t is used to express beliefs.

[14] The model-world pair is just a shorthand for $(\mathtt{M,KB,B}, w)$. As it will be clear in the sequence satisfiability depends on the model $\mathtt{M}$ of the knowledge base $\mathtt{KB}$, the knowledge base itself, the belief time relation $\mathtt{B}$ and a possible world w. For brevity we will use only the pair $(\mathtt{M}, w)$ but will mean the quadruple $(\mathtt{M,KB,B}, w)$.

For well-formed formulae that involve quantifiers, we distinguish two cases: one where the variable bound by the quantifier are non-temporal, and one where the variable ranges over a set of time intervals. In the first case we have for a variable assignment i mapping variables in elements of $\mathtt{S(C)}$:

$(\mathtt{M}, w) \models (Forall\ x/C)\ \alpha$ iff for every variable assignment i' mapping variables in elements of $\mathtt{S(C)}$ and differing from i possibly on the value of $x, (\mathtt{M}, w) \models \alpha$

$(\mathtt{M}, w) \models (Exists\ x/C)\ \alpha$ iff for some variable assignment i' mapping variables in elements of $\mathbf{S}(C)$ and differing from i possibly on the value of $x, (\mathtt{M}, w) \models \alpha$

In the second case :

$(\mathtt{M}, w) \models (Forall\ t/\mathtt{TimeInterval})\ \alpha$ iff $\forall w' \in \mathtt{W}$ such that wRw' or $w'Rw$, $(\mathtt{M}, w') \models \alpha$

$(\mathtt{M}, w) \models (Exists\ t/\mathtt{TimeInterval})\ \alpha$ iff $\exists w' \in \mathtt{W}$ such that wRw' or $w'Rw$,

and $(\mathtt{M}, w') \models \alpha$

where `TimeInterval` is the built-in class comprising all time intervals.

5 A Functional Specification of Telos Knowledge Bases

In this section `Telos` knowledge bases will be specified functionally, i.e., in terms of the operations one is allowed to perform on them, as in e.g. (Levesque 1984) and KRYPTON (Brachman, Fikes and Levesque 1983). We will show what the result of any of the operations on the initial or any consecutive knowledge base is, in terms of the already defined semantics of the basic framework constructs and all sorts of propositions a `Telos` knowledge base may contain. With this knowledge-level analysis we can exactly specify what a `Telos` knowledge base can be asked or told about the domain of discourse.

The operations constituting the interaction mechanism of `Telos` knowledge bases are the operations `TELL, UNTELL, RETELL, RETRIEVE` and `ASK`. A `Telos KB` can be updated through `TELL, UNTELL` and `RETELL` and can be queried through `ASK` and `RETRIEVE`. All these operations, operate on sets of propositions (`KB`'s). From a semantic point of view these sets of propositions are sets of possible worlds. In this section we will specify the result of each of the operations in terms of the compatible possible worlds that comprise the resulting `KB`.

5.1 The TELL Operation

`TELL` can be used to enter a new object in the knowledge base or to append new attributes to already defined objects. Below we depict the full form a `TELL` command can have when issued to an existing knowledge base. Its arguments are a knowledge base `KB` and a definition ϕ of a `Telos` object. It returns a new knowledge base `KB`' that contains the definition of the new object and all its associated relationships that have been defined by `TELL`.

In the specification of the `TELL` command below, α stands for `TOKEN` or `CLASS`, β is any identifier, δ and ϵ are classes, ζ is an identifier, ι and λ are attributes, κ is an attribute value (i.e. a proposition or an assertion), μ is an assertion, $\theta, \theta_1, \theta_2, \theta_3, \theta_4$ are temporal relations and $\tau, \tau_1, \tau_2, \tau_3, \tau_4$ are time constants.

`TELL` $\alpha\ \beta\ \ (\theta\ \tau)$

```
        IN
              δ (θ1 τ1)
        ISA
              ϵ (θ2 τ2)
        WITH
              ζ
              ι : κ (θ3 τ3)
              λ : $ μ$ (θ4 τ4)
END
```

Assume ϕ denotes a complete definition of a `Telos` object. Then, $\sigma[\![\texttt{TELL(KB},\phi)]\!] = \sigma[\![\texttt{KB}]\!] \cap \texttt{W}_\phi$, where $\texttt{W}_\phi \subseteq \texttt{W}$ and is defined as the set of possible worlds w for which the following set of propositions is satisfied for any model `M` of the knowledge base whose set of possible worlds is a subset (not necessarily proper) of the set of possible worlds of the model of the initial knowledge base:
$\{\texttt{prop}(\beta,\beta,l,\beta,t_1)$, $\texttt{instanceOf}(\beta,\texttt{Individual},t_2,t_3)$,
if $\alpha = \texttt{TOKEN}$, $\texttt{instanceOf}(\beta,\texttt{IndividualClass},t_2,t_3)$, if $\alpha = \texttt{CLASS}$, $(t_1\ \theta\ \tau)$, $(t_2\ \theta\ \tau)$,
$(t_3\ \texttt{costarts systime})$, $\texttt{instanceOf}(\beta,\delta,t_5,t_6)$, $(t_5\ \theta_1\ \tau_1)$, $(t_6\ \texttt{costarts systime})$, $\texttt{isA}(\beta,\epsilon,t_7,t_8)$,
$(t_7\ \theta_2\ \tau_2)$, $(t_8\ \texttt{costarts systime})$, $\texttt{prop}(p_1,\beta,\iota,\kappa,t_9)$,
$(\forall x,y,z\ /\texttt{Proposition})(\forall u\ /\texttt{TimeInterval})\ \texttt{prop}(x,y,\zeta,z,u)\ \mathit{and}$
$(\exists t_1,t_2\ /\texttt{TimeInterval})\ \texttt{instanceOf}(\beta,y,t_1,t_2) \Rightarrow \texttt{instanceOf}(p_1,x,t_9,t_{10})$,
$(t_9\ \theta_3\ \tau_3)$, $(t_{10}\ \texttt{costarts systime})\ \}$

The above formulae specify the result of a `TELL` operation on a `Telos` knowledge base. The result is a new knowledge base where the information supplied by `TELL` has been added and its semantics consists of the set of possible worlds (a subset of the set `W` of all possible worlds) which are compatible with the new piece of information. In section 2.1 the object `krypton` was defined. Assume this `TELL` command is issued to the knowledge base. The semantics of the new knowledge base results from the intersection of the set of possible worlds constituting the semantics of the previous knowledge base with the set of all possible worlds in which `krypton` is a token proposition, an instance of the class paper, the authors are `Brachman, Fikes` and `Levesque` who started working on the paper at `1986/10, 1987/1` and `1986/10` respectively, the title is `A Hybrid Representation System`, the paper is dated `1986/12` and is classified in the subject category `kr`. These latter possible worlds are such that their time indices fall within the specified time intervals. In addition the propositions defining the properties of the new object are believed by the system for time intervals that depend on the time the operation took place. Of course, the basic framework axioms as well as the constraints on `InstanceOf` and `IsA` relations must be preserved in any knowledge base.

Assuming that they hold in the previous knowledge base, it is not hard to see that the semantics defined preserve both the axioms and the constraints in the knowledge base resulting from the `TELL` operation.

5.2 The Commands UNTELL and RETELL

`UNTELL` can be used to inform the knowledge base that a specific relationship of an object that has been explicitly defined (i.e. a relationship defined by a `TELL` command) does not hold for a certain period of time. What is important to note is the fact that no deletion of propositions in the knowledge base occurs when an `UNTELL` command is executed. What is changed is the beliefs of the systems about the propositions defining the relationships that `UNTELL` specifies. `UNTELL`'s arguments are a knowledge base `KB` and a (not-necessarily complete) definition ϕ of the object and/or its associated relationships that are to be `UNTELL`-ed. The result is a new knowledge base $\mathtt{KB}'$ which will reflect the system's change of beliefs.

The following simple example illustrates how the knowledge base resulting from an `UNTELL` command can be specified at the knowledge level. In this example, an explicitly defined relationship of an object is `UNTELL`-ed.

Consider the following `TELL` command defining *Ron*.

```
TELL  TOKEN Ron
      WITH
            name
                  :'Ron Brachman'
            affiliation
                  :'UofT'
            address
            home-address : r-addr (at  1986/10/01 .. *)
END
```

Assume now the following `UNTELL` command is executed.

```
UNTELL  TOKEN Ron
        WITH
              address
              home-address :  r-addr (at  1988 .. *)
END
```

The beliefs that the system used to hold about the address of Ron will remain unchanged for the time interval (1986/10/01..1987/12/31). However they will be revised for (1988..$*$). As was mentioned before, `UNTELL` results in no deletion of any propositions in the knowledge base. On the contrary, new propositions are added to it reflecting the fact that the systems beliefs have changed. Continuing our example, the propositions

$p_4 :=< p_1, instanceOf,\ P, t_2' >$

$p_5 :=< p_4, instanceOf,\ \mathtt{InstanceOf}, t_5 >$

will be added to the knowledge base as a result of the execution of the above `UNTELL` command.

In addition, for any model `M` (as before) and any possible world $w \in \sigma[\![\mathtt{KB}']\!]$, where $\mathtt{KB}'$ is the resulting knowledge base, the folowing will hold :

$(\mathtt{M}, w) \models$ $\mathtt{instanceOf}(p_1, P, t_2')$
$(\mathtt{M}, w) \models$ $\mathtt{instanceOf}(p_4, \mathtt{InstanceOf}, t_5)$
$(\mathtt{M}, w) \models$ $(t_3\ \mathtt{coends\ systime})$
$(\mathtt{M}, w) \models$ $(t_2'\ \mathtt{costarts}\ t_2)$
$(\mathtt{M}, w) \models$ $(t_2'\ \mathtt{coends}\ 1987)$
$(\mathtt{M}, w) \models$ $(t_5\ \mathtt{costarts\ systime})$
and $\mathtt{B}(p_5, t_5)$. Hence, the set of possible worlds in the new knowledge base will result by intersecting the previous knowledge base with the set of possible worlds in which the above hold.

RETELL can be used to inform the knowledge base that a specific relationship that has been explicitly created for a certain object does not hold for a certain time period and some other relationship holds in its place. This relationship can be an instantiation, specialization or attribute relationship. The new relationship introduced by RETELL must be of the same kind as the one that RETELL specifies as a non-valid one for the specific time interval. The result of a RETELL command is a new knowledge base in which the relationships appearing in the IN, ISA and WITH clauses have been UNTELL-ed and the relationship introduced has been TELL-ed.

The semantics of RETELL can now be defined compositionally based on the already defined semantics of TELL and UNTELL. The possible worlds in the semantics of the resulting knowledge base are those that would result if the corresponding UNTELL command had been issued to the initial knowledge base, intersected with the set of possible worlds in the semantics of the knowledge base that would result from the execution of a TELL command specifying the new relations on the knowledge base resulting from UNTELL.

$$\sigma[\![\mathtt{RETELL}(\mathtt{KB}, \phi)]\!] = \sigma[\![\mathtt{UNTELL}(\mathtt{KB}, \phi')]\!] \cap \sigma[\![\mathtt{TELL}(\mathtt{KB}', \phi'')]\!]$$

where ϕ is the full form of the RETELL argument, ϕ' is the corresponding argument in the UNTELL command (that specifies the relationships that are to be marked as non-valid, ϕ'' is the argument of a TELL command made up from the parts of ϕ that introduce the new relations, and $\mathtt{KB}' = \mathtt{UNTELL}(\mathtt{KB}, \phi')$.

5.3 The Commands ASK and RETRIEVE

Queries to the knowledge base can be issued through the operations ASK and RETRIEVE. Their syntax is very similar and they only differ in that ASK answers queries by taking the deductive rules into account as well, whereas RETRIEVE considers only the terminological information in the knowledge base. The general form of a ASK command is as depicted below.

ASK $x_1/s_1, \ldots, x_n/s_n\ : \phi$
ON t_1
AS OF t_2

ϕ is a well-formed formula of the assertion language of Telos, $x_1, \ldots, x_n$ are variables occurring free in ϕ and $s_1, \ldots, s_n$ are sets denoting the ranges of $x_1, \ldots, x_n$ respectively. t_1 and t_2 are time intervals denoting the history and belief times according to which we are interested in answering the query. If the part $x_1/s_1, \ldots, x_n/s_n$ is present, then the result will be the set of all possible substitutions for $x_1, \ldots, x_n$ for which ϕ comes out true. Otherwise, the result is one of the answers `yes`, `no` or `unknown`. For a model `M` of the knowledge base we have:

$\texttt{ASK}(\texttt{KB}, \phi) =$ `yes` if and only if $\forall w \in \delta(t_1)$, $(\texttt{M}, w) \models \phi$, and for every proposition $p \in \texttt{KB}$ that represents some primitive or generic subformula of ϕ, it is the case that $\texttt{B}(p, t_2)$, or there exists a set of deductive rules Δ in the knowledge base such that $(\texttt{KB}, \Delta) \models \phi$

$\texttt{ASK}(\texttt{KB}, \phi) =$ `no` if and only if $\forall w \in \delta(t_1)$, $(\texttt{M}, w) \models \neg\phi$, and for every proposition $p \in \texttt{KB}$ that represents some primitive or generic subformula of $\neg\phi$, it is the case that $\texttt{B}(p, t_2)$, or there exists a set of deductive rules Δ in the knowledge base such that $(\texttt{KB}, \Delta) \models \neg\phi$

$\texttt{ASK}(\texttt{KB}, \phi) =$ `unknown` if and only if it is not the case that $\forall w \in \delta(t_1)$, $(\texttt{M}, w) \models \phi$ or $(\texttt{M}, w) \models \neg\phi$, or there is no set of deductive rules Δ that entails ϕ, or the system has no beliefs concerning primitive or generic subformulae of ϕ.

Entailment of a well-formed formula ϕ by a set of deductive rules Δ in a knowledge base `KB` is defined as:

$(\texttt{KB}, \Delta) \models \phi$ if and only if for all models $\mathbf{M}$ of the knowledge base and possible worlds w such that $(\texttt{M}, w) \models d$, $\forall d \in \Delta$ it is also the case that $(\texttt{M}, w) \models \phi$.

Returning to our example, assume the following definition and queries are issued to the knowledge base:

```
TELL  TOKEN Hector
      WITH
            name
                  :'Hector Levesque'
            affiliation
                  :'UofT'
            address
            home-address : h-addr (at  1980 .. *)
END
```

`ASK` : *Ron* ˆhome-address = r-addr
`ON` 1987/09
and
`RETRIEVE` : *Hector* ˆhome-address = r-addr

The answer to the first query[15] will be `yes`, while that to the second query will be `no`. The reason is that, for a model `M` of the knowledge base, for all possible worlds $w \in \delta(1980/10/01..*)$, $(\texttt{M}, w) \models$ (*Ron* ˆhome-address = r-addr). while there is no possible world in $\delta(1980..*)$ for which $(\texttt{M}, w) \models$ (*Hector* ˆhome-address = r-addr). Also, for the first query, it is believed by the system that *Ron* ˆhome-address = r-addr for a time interval t during `Now`.

Assume now that the following definition has been added to the knowledge base

[15] Since no deductive rules are involved, the answers would be the same if `RETRIEVE` were used in place of `ASK`.

```
TELL  TOKEN UofT
      IN Employer
      WITH
            emp-name
                        :'University of Toronto'
            emp-address
                        : UofT-addr
END
```

together with a deductive rule asserting that every address of the employer of an author is an address of the author himself. Then the answer to the query

`ASK` : UofT-addr `MemberOf` Ron.address

will be `yes`, while if the same query were formulated with `RETRIEVE` the answer would be `no`, because of the presence of the deductive rule.

6 Consistency and Completeness of Telos

In this chapter we will investigate the consistency and completeness of the formalism presented in the previous section. If consistency and completeness can be proven, then any implementation that respects the defined semantics and the knowledge-level specification of `Telos` knowledge bases, can be proven correct.

We will start by defining the inference rules to be used in this attempt. The axiomatization has been presented along with the examination of the specific constructs of `Telos` for which a semantics has been defined. Then, Soundness, Completeness - for a restricted case - and Consistency of the formalism will be proven. In a nutshell, what we want to prove in this chapter is the following:

`The Completeness Theorem of Telos:`$\texttt{ASK}(\texttt{KB}, \phi) =$ `yes` iff $\texttt{KB} \vdash \phi$

where ϕ is any first-order sentence of the assertion language and $\vdash$ is a derivability relation that will be defined in the sections to follow. In other words, we want the knowledge base to provide the answer `yes` to a query if and only if the query (in the form of a first-order sentence) can be derived from the knowledge base, and, conversely, for any sentence of the assertion language that can be derived from the knowledge base, `ASK` must reply `yes` when given the knowledge base and the sentence as arguments. We are using this weak version of completeness since we allow the operation `ASK` to be incomplete. As was shown in the previous chapter, `ASK` returns `unknown` whenever neither ϕ nor $\neg\phi$ are satisfied in the knowledge base.

`ASK` has been specified formally in the previous chapter. A `yes` answer to an `ASK` command is equivalent to having a model satisfying the query for all possible worlds in the semantics of the knowledge base $\sigma[\![\texttt{KB}]\!]$.

6.1 Some Remarks on the Axiomatization of Telos

The axiomatization of `Telos` is described in detail in (Stanley 1986) and (Koubarakis et al. 1989) . The basic framework axioms and the time axioms are presented there. In this section, we will comment on this axiomatization.

It can be easily seen that `Telos` is not finitely axiomatizable in first order logic. The infinite dimension created by the instantiation relationship is governed by axioms stating the existence of the built-in classes of `Telos` at each level of the dimension and the relationships that hold among them. The built-in `Telos` class at each level of the instantiation dimension is an instance of the class at the next level of the dimension, and all of them are instances of the built-in class `Proposition`.

In addition, if we were to express axioms preventing the creation of paradoxes like the Russell's paradox, then these would have to be in the form of second order axiom schemata. We cannot express in ordinary first order logic that a class cannot be an instance of itself. However, in a language like the one the semantics were expressed, such axioms can be stated. For instance, the axiom preventing a class from being an instance of itself could be written in the form of a constraint imposed on the knowledge base, as:

$\forall p, q \in \texttt{KB}$ *such that* $\gamma(p), \gamma(q) \in \texttt{S(Class)}$,

$\exists t$ *such that* $\forall w \in \delta(t) < \gamma(p), w > \in C(q) \Rightarrow \gamma(p) \neq \gamma(q)$

If $p = q$, then $\gamma(p) = \gamma(q)$. Hence, $\gamma(p) \neq \gamma(q)$ implies $p \neq q$.

The above constraint prevents Russell's paradox by disallowing a class from being an instance of itself. If a `Telos` definition that, explicitly or implicitly, results in a class being an instance of itself is entered in the knowledge base, the constraint will be violated and the definition will not be accepted.

6.2 Inference Rules

In this section the inference rules that we will employ in proving soundness and adequacy of `Telos` will be introduced. The inference rules will be formulae preserving the validity of sentences in any model `M`.

As far as the terminological part of `Telos` is concerned, the inference rules consist of:

(a) Transitivity of the *IsA* relationship

(b) Default rules about the persistence of belief time intervals

These rules can be expressed formally as:

(a) $(\forall p, q, r)(\forall t, t1, t2, t3)(\gamma(p), \gamma(q), \gamma(r) \in \texttt{S(Class)}$

$\wedge\ \texttt{isA}(p, q, t1, t)\ \wedge\ \texttt{isA}(q, r, t2, t)\ \wedge\ (t3\ \textit{is the common interval of}\ t1, t2)$

$\Longrightarrow \texttt{isA}(p, r, t3, t)$[16]

Persistence of beliefs is formally expressed by the following default rule (following Reiter's default logic (Reiter 1980)):

[16] This rule could be written in a more general form, in which the belief time intervals of `isA` propositions are not identical, but merely overlapping, or preferably, one is included in the other since they are considered to be persistent.

(b)

$$\frac{(\exists p, t0)(\texttt{instanceOf}(p, \texttt{InstanceOf}, t, t0): \ M(t \ meets \ +\infty)}{(t \ meets \ +\infty)}$$

That is, if it is consistent to assume that a belief time interval is right-infinite, then we can assume so.

As far as the assertional component is concerned, we adopt a natural deduction system for first-order logic - as e.g. in (Cohen 1987) - with the standard rules for *Elimination* and *Introduction of conjunction*, *Elimination* and *Introduction of material implication*, $\perp$, and *Universal Instantiation* and *Generalization*. We will also assume that the axioms of first-order logic are known.[17]

We will also employ two "meta-rules" that are needed for the purposes of proving completeness. These rules follow from Allen's (Allen 1984) axiomatization of time. These rules could be expressed using the predicate "*Holds*" which is used to specify that a certain proposition holds for a specific time interval. Hence, in terms of our notation, for a knowledge base `KB`, $Holds(\phi, t)$ if and only if for a model `M` of `KB` $(M, w) \models \phi \forall w \in \delta(t)$. For simplicity, the predicate "Holds" is not used in the following rules.

[TI $\forall$ **]**: From $\forall w \in \delta(\texttt{Alltime})$ $(\texttt{M}, w) \models \phi$, for a given model `M`, infer , $(\texttt{M}, w) \models (\forall t/\texttt{TimeInterval})\phi$

[Ex]: If ϕ holds for the consecutive time intervals t_1 and t_2, then it also holds for the time interval that is defined from the startpoint of t_1 and the endpoint of t_2

The applicability of the above metarules is obvious. If a sentence holds for the time interval `Alltime` then we can assume that it holds for every time interval, or in the terminology of our possible-worlds semantics, if it holds for every possible world, then it holds for every time instance. The last rule, which, due to lack of a better name, has been called the rule of "extension to consecutive time intervals", intends to capture the intuition that if a sentence holds for two consecutive time intervals, it also holds for their union.

It can also be shown (see (Cohen 1987)) that rules for elimination and introduction of disjunction and of elimination and introduction of existential quantification can be derived from the existing rules.

We now define , in a standard way, the *derivability relation* $\vdash$, so that $\Gamma \vdash \alpha$ means that the closed formula α can be derived from the sentences in Γ (in our case a knowledge base `KB`), and the axioms by applying the inference rules.[18]

[17] According to Levesque (1984b), , this might be called the assumption of *competence*.

[18] This is the standard way of defining derivability. In `Telos` knowledge bases however, deductive rules may also pe present allowing us to derive new facts. We will consider that deductive rules appear in a knowledge base `KB` in the form of closed formulae. In fact, the rule for *Universal Instantiation* captures the applicability of a deductive rule in the knowledge base.

6.3 Soundness, Consistency and Completeness

6.3.1 Soundness

Soundness is expressed by the following theorem for a given `Telos` knowledge base `KB`:

`Soundness Theorem`: For every closed formula $\alpha, \mathtt{KB} \vdash \alpha \Rightarrow \mathtt{ASK(KB,}\alpha\mathtt{)} = \mathtt{yes}$

In other words, everything that is derivable from a `Telos` knowledge base, is satisfiable in any model `M` and any possible-world w that satisfy all closed formulae in the knowledge base.

The proof of the theorem is straightforward. For the assertional part of `Telos` the inference rules we use to derive new information are truth-preserving rules of a natural deduction system. It suffices that, to show that the default rule for the persistence of beliefs, the transitivity of *IsA* and the rule for introduction of temporal universal quantification do not result in deriving sentences that are not satisfiable in the given model `M`.

(a) Assume that the propositions expressing $\mathtt{isA}(p,q,t1,t)$ and $\mathtt{isA}(q,r,t2,t)$ are in the knowledge base and that there exists a common time interval $t3$ for the time intervals $t1$ and $t2$. Then, for any model `M` and any possible world w for which $(\mathtt{M},w) \models \mathtt{isA}(p,q,t1,t)$ and $(\mathtt{M},w) \models \mathtt{isA}(q,r,t2,t)$, it is also the case that $(\mathtt{M},w) \models \mathtt{isA}(p,r,t3,t)$. This is because

$\forall w \in \delta(t1) \ < \gamma(p), \gamma(q), \gamma(t1), \gamma(t) > \in V(\mathtt{isA})$

$\forall w \in \delta(t2) \ < \gamma(q), \gamma(r), \gamma(t2), \gamma(t) > \in V(\mathtt{isA})$

Hence,

$\forall w \in \delta(t1) \cap \delta(t2)$ both the above tuples are in the extension of `isA`, and both these facts are believed for the same time interval t. From the *Specialization Postulate* all instances of p for $t1$ will be instances of q for $t1$ and all instances of q for $t2$ will be instances of r for $t2$. Hence all instances of p will be instances of r for the common interval $t3$ of $t1$ and $t2$. Moreover, if $(\mathtt{M},w) \not\models \mathtt{isA}(p,r,t3,t)$ the *Specialization Postulate* is violated. Hence $(\mathtt{M},w) \models \mathtt{isA}(p,r,t3,t)$, and the corresponding proposition is believed for the time interval t.

(b) The inference rule about the persistence of belief time intervals, simply states that propositions that are believed continue to be believed until an `UNTELL` or `RETELL` operation informs the knowledge base otherwise. Hence, if there is no time interval t' such that $< \gamma(p), t' > \in \mathtt{B}$ and $(\mathtt{M},w) \not\models (t \ \mathtt{coends\ systime})$, then $(\mathtt{M},w) \models (t \ \mathtt{meets}\ +\infty)$.

(c) The result is obvious for the rule of temporal universal quantification since , $(\mathtt{M},w) \models (\forall t/\mathtt{TimeInterval})\phi$ and $(\mathtt{M},w) \models \phi \ \forall w \in \delta(\mathtt{Alltime})$ are equivalent expressions.

(d) It is also trivial to see that the rule of extension to consecutive time intervals is truth-preserving. □

6.3.2 Consistency

Having established Soundness of our formalism, Consistency can be established by means of the following result:

`Consistency Theorem:` A `Telos` knowledge base is consistent if it has a model.

`Proof:` Suppose a `Telos` knowledge base `KB` has a model `M`. Then, for all possible worlds $w \in \sigma[\![\mathtt{KB}]\!]$ it is the case that no contradiction is satisfied by this model-world pair ($(\mathtt{M},w) \not\models$

$false$). From the Soundness Theorem it follows that no contradiction can be derived from the knowledge base ($\mathtt{KB} \nvdash false$), which proves the consistency of any `Telos` knowledge base. □

6.3.3 Completeness

The following theorem needs to be proven in order to establish the completeness of `Telos` with respect to the defined semantics and the functional specification of `Telos` knowledge bases.

`Completeness Theorem`: For any closed formula α, if $\mathtt{ASK}(\mathtt{KB}, \alpha) =$ `yes`, then $\mathtt{KB} \vdash \alpha$

`Proof` : In the previous chapter a specification of the `ASK` operation was defined in terms of the satisfiability of the formula that constitutes the query on the knowledge base and the possible worlds that satisfy the temporal constraints.

Assume that the following query is posed to a knowledge base `KB`:

`ASK` (`KB`, ϕ)
`ON` t_1
`AS OF` t_2

where ϕ is a well-formed formula of the assertion language.

We restrict ϕ to allow temporal quantification over definite time intervals only. A "`yes`"-answer to the above query is equivalent to a "`yes`" - answer to all queries of the form `ASK(KB,` ϕ'), where each ϕ' results from ϕ by eliminating universal or existential quantification of temporal variables and substituting each occurrence of a history time interval with an instantaneous time interval that consists of the time index of a possible world in the set of possible worlds whose indices satisfy the temporal constraints of the query. Since we only allow non-infinite time intervals this set of possible worlds will be finite. Also recall that all possible relations between two time intervals can be written in terms of the union and intersection of the corresponding sets of possible worlds. Hence, the time indices (instantaneous time intervals) for which we have to check for the satisfiability of ϕ' is uniquely defined.

Having eliminated temporal quantification, satisfiability for model-world pairs $(\mathtt{M}, w)$ is now equivalent to treating time as static, i.e., as an ordinary attribute in atomic formulae of the language. With time eliminated from quantification the language is equivalent to that of ordinary first-order logic.

Completeness (or Adequacy as is called by certain authors) results of First-Order Logic, as in (Shoenfield 1967), guarantee that whenever a first-order formula is valid , it can actually be derived from the axioms with the use of deductive rules. In addition a many-sorted logic is equivalent to First Order Logic (Enderton 1972) . Hence, in our case, for each such ϕ' that is satisfied for a model-world pair $(\mathtt{M}, w)$ that satisfies all closed formulae of the knowledge base, it is the case that $\mathtt{KB} \vdash \phi'$. By the rule of introduction of conjunction, it follows that

$\mathtt{KB} \vdash \bigwedge_i \phi'_i$,

where ϕ'_i is the formula that results from ϕ by the substitution described above and corresponds to the time index i. Those indices correspond to all time points that belong to the time interval(s) that express the temporal constraint. From the rule of extension to consecutive time intervals, it follows that $\mathtt{KB} \vdash \phi$, if one time interval specifies the temporal constraint, and in the case that more than one intervals specify this constraint, the rules of extension to consecutive time intervals and of introduction of conjunction can be used to infer $\texttt{KB} \vdash \phi$.

Hence, if $\texttt{ASK}(\texttt{KB}, \phi) = \texttt{yes}$, then $\texttt{KB} \vdash \phi$. □

In the above proof, belief time was not mentioned at all. The beliefs of the system are consulted when satisfiability for a model-world pair is tested. From the specification of `ASK` in the previous chapter follows that `ASK` replies "`yes`" exactly when what is satisfied for the appropriate model-world pairs is also believed by the system for the proper time intervals with the underlying assumption that, when a proposition is believed for a time interval t, it is also believed for all subintervals of t. Hence, when the transformation of the query to a set of queries preserves the systems beliefs. A key point in this proof is that the number of model-world pairs with respect to which satisfiability is tested, is finite. This is the reason that queries are restricted over non-infinite time intervals.

7 Conclusions and Directions for Further Research

7.1 Results

The problems dealt with in this paper, along with the proposed solutions can be summarized as follows:

- An ontology for `Telos` allowing abstract concepts and objects existing at other times to be treated as first-class entities. The proposed ontology was based on the property of existence whose treatment was inspired by the tight integration of time in `Telos`. Following Woods' (1975) and Hirst's (1989) guidelines on the ontological requirements for a knowledge representation system, this ontology was defined in an attempt to make the system more suitable for real-world applications.

- A formal semantics for `Telos` knowledge bases that accounts for the several kinds of entities the proposed ontology consists of and is intuitive enough to capture the temporal dimension of `Telos` and its object-centered representational framework. The semantics is based on a possible-worlds model that includes a linear and non-dense temporal structure.

- A functional specification of `Telos` knowledge bases at the knowledge level was provided, allowing for the characterization of the result of the interaction with a knowledge base by means of operations querying and updating a knowledge base and independently of any implementation. The operations `ASK, RETRIEVE` for querying the knowledge base and `TELL, RETELL` and `UNTELL` for updating the knowledge base have been characterized functionally in terms of the set of possible worlds (epistemic state) that the semantics of the resulting knowledge base consists of.

- Soundness, Completeness and Consistency results (for a subset of `Telos` at least) that complete the semantic treatment of the language and will constitute, along with the previous results, a testbed for verifying the correctness of any implementation.

In the course of this formal account other problems or weaknesses of the language were encountered as well. For instance, it was shown that `Telos` is not finitely axiomatizable in first-order logic and that the existing framework does not prevent the creation of paradoxes like Russell's paradox. The semantics provided for `Telos` in this thesis appears to be more intuitive than the semantics given for `Telos`'s predecessor `CML` in (Stanley 1986) , where the language was mapped to an ordinary first-order logic. In (Koubarakis 1988) language expressions were mapped to `Telos` knowledge bases. The semantics defined in this thesis assigns a meaning to `Telos` knowledge base structures and interprets propositions in the domain of discourse.

7.2 Some General Remarks on Telos

`Telos` is a sophisticated knowledge representation language designed to support the development of information systems, especially during the requirements modeling phase. Its contribution lies in the adaptation of ideas from several areas, such as Knowledge Representation, Deductive Databases and Requirements Modeling.

The semantics provided was mainly influenced from work in the area of Modal and Temporal Logics, Knowledge Representation, Linguistics and Philosophy. Certainly several problems and questions remain unresolved, but steps towards achieving epistemological adequacy have been made.

`Telos` can be used to build knowledge bases that will support all activities in the process of developing information systems. Its uniform object-centered framework along with a number of special features have enhanced its ability for maintaining knowledge bases.

Several implementations of `Telos` have been or are being carried out. Experiences in implementing `Telos` at the University of Toronto are described in (Topaloglou and Koubarakis 1989) . A Prolog implementation was developed for the ESPRIT project LOKI (Binot et al. 1988) for supporting natural language interaction with knowledge bases. Another Prolog implementation was used for modeling the design history of information systems within the DAIDA ESPRIT Project (Jarke et al. 1990) . Apparently, the most complete Prolog-based implementation of `Telos` is the one for ConceptBase (Jarke et al. 1988). Most recently, `Telos` has been used in the ESPRIT II Project ITHACA (Constantopoulos et al. 1989) . A Lisp-based `Telos` implementation, by the name KNOWBEL (Kramer et al. 1991) , has been used for developing an alarm filtering expert system for a nuclear power plant. Finally, a Smalltalk implementation is under way (Sobiesiak and Mylopoulos 1991) for supporting Authoring-in-the-Large for Hypertext Documents. A detailed investigation of the applications of `Telos` can be found in (Mylopoulos et al. 1990) .

7.3 Future Research

Future considerations include the addition of modalities in the language, such as Levesque's K operator (Levesque 1984) , that will enable statements about the knowledge base. Of interest would be an operator allowing statements about the domain of discourse, an inverse-K operator or an *actuality* operator in the spirit of (Meyer and Hoek 1989) . The investigation of the epistemic nature of `Telos` and its potential applicability to reasoning about action and plans are also areas where research can be focused.

8 Acknowledgements

The author would like to thank Professor John Mylopoulos for his ideas and his insightful guidance towards the completion of this work. Professor Hector Levesque provided invaluable comments on earlier versions of this work. Financial support was provided by the University of Toronto and ITRC.

A Appendix

A.1 The Possible-Worlds Model

The possible-worlds model is defined as a 7-tuple

- $\mathtt{M} = < \mathtt{W}, \mathtt{D}, \mathtt{Q}, \mathtt{T}, \mathtt{<}, \delta, \mathtt{R} >$

where:

$\mathtt{W}$: set of all possible worlds

$\mathtt{D}$: non-empty set of individuals comprising the domain of discourse.

$\mathtt{Q}$: function assigning to each world $\mathtt{w}_i \in \mathtt{W}$ a certain subset $\mathtt{D_i}$ of $\mathtt{D}$ as its own peculiar domain.

$\mathtt{T}$, $\mathtt{<}$: a denumerable set of time points equipped with a binary order of precedence $\mathtt{<}$, which is a linear order.

$\delta : \mathtt{I} \rightarrow 2^{\mathtt{W}}$: a function associating possible worlds with time intervals.

$\mathtt{R}$: accessibility relation between possible worlds.

- The set $\mathtt{I}$ of time intervals is defined as $\mathtt{I} = \{(t_1, t_2)/(t_1, t_2) \in \mathtt{T} \times \mathtt{T}, t_1 \leq t_2\} \cup \{\mathtt{Alltime}\} \cup \{(t, \infty)/t \in \mathtt{T}\} \cup \{(-\inf ty, t)/t \in \mathtt{T}\}$.[19]
- The set $\mathtt{P}$ of eternal objects can now be defined as $\mathtt{P} = \mathtt{D} - (\bigcup_{\mathtt{w}_i \in \mathtt{W}} \mathtt{D}_i)$.
- The time model adopted is a linear (non-branching), infinite and non-dense model. It is characterized by the following properties:

`linearity` : $\forall x, y((x < y) \vee (y < x) \vee (x = y))$

`irreflexivity` : $\forall x(\neg(x < x))$

`discreteness` : $\forall x, y((x < y) \Rightarrow \exists z((x < z) \wedge \neg \exists w((x < w) \wedge (w < z))))$

`succession` : $\forall x \exists y(x < y)$

[19] The $\leq$ relation can be defined from $\mathtt{<}$ and equality in the usual manner: $\forall x, y : x \leq y$ iff $(x < y) \vee (x = y)$.

Function δ satisfies the following properties:

- If $i_1, i_2 \in \mathtt{I}$ with $i_1 \sqsubseteq i_2$, then $\delta(i_1) \subseteq \delta(i_2)$[20]

- $\delta(\mathtt{Alltime}) = \mathtt{W}$

- If $t \in \mathtt{T}$, $\delta(t, \infty) = \mathtt{W} - \bigcup_{t' \in \mathtt{T} - \{t'' | t'' < t\}} \delta(t, t')$

• $\mathtt{B} \subseteq \mathtt{D} \times \mathtt{I}$ is a relation representing when (for which time interval) a proposition in the knowledge base is believed by the system.

• `S` : a set of partially overlapping sorts, containing a sort for each built-in and user-defined class of `Telos`. If `C` is a class, `S(C)` will denote the sort that corresponds to class `C`.

A.2 The semantic functions

• `V` is an extension function mapping n-place predicate symbols to sets of ordered (n+1)-tuples of elements of `D` and `W`.

• $\mathtt{V}(\phi) = \{< u_1, u_2, ..., u_n, \mathtt{w} > / u_i \in \mathtt{D}, i = 1, .., n \ and \ \mathtt{w} \in \mathtt{W}\}$.

• `E` is an extension function mapping `Telos` classes to the corresponding relations over the domain of discourse `D` in the same manner as `V`.

• `C` is an extension function mapping defined classes to relations over `D` and `W`, in the same manner as `E`.

• γ is a function mapping `Telos` propositions to their corresponding individuals in the domain of discourse. That is, if $p \in \mathtt{KB}$,then $\gamma(p) \in \mathtt{D}$.

• σ will be the function assigning meaning to propositions or sets of propositions (knowledge bases).

A.3 Semantics

For the initial knowledge base $\mathtt{KB}_0$, $\sigma[\![\mathtt{KB}_0]\!] = \mathtt{W}$

• If `p` is an individual token proposition, then $\sigma[\![\mathtt{p}]\!] = \gamma(p)$, and $\gamma(p) \in \mathtt{D}$ and if $p \in \mathtt{KB}$, then $\sigma[\![\mathtt{p} |\ when(\mathtt{p}) = i]\!] = \gamma(\mathtt{p})$ and $\forall w \in \sigma[\![\mathtt{KB}]\!]\ \gamma(\mathtt{p}) \in \mathtt{Q}(w)$ iff $w \in \delta(i)$.

• If `p` is an attribute token proposition, then $\sigma[\![\mathtt{p} |\ when(\mathtt{p}) = i]\!] \subseteq \mathtt{D} \times \mathtt{D} \times \mathtt{W}$, and $\sigma[\![\mathtt{p} | when(\mathtt{p}) = i]\!] = \{< o_1, o_2, w > \mid o_1, o_2 \in \mathtt{Q}(w) \cup \mathtt{P}\ o_1 = \gamma(from(\mathtt{p}))\ , o_2 = \gamma(to(\mathtt{p}))$ and $w \in \delta(i)\}$

• If `p` is an individual simple class, then the extension of `p` is `E(p)`.

• If `p` is an attribute simple class, then $\sigma[\![\mathtt{p} | when(\mathtt{p}) = i]\!] = \{< o_1, o_2, w > | o_1, o_2 \in \mathtt{P},\ o_1 = \gamma(from(\mathtt{p})), o_2 = \gamma(to(\mathtt{p})),$ $< o_1, w > \in \mathtt{E}(from(\mathtt{p})), < o_2, w > \in \mathtt{E}(to(p))$ and $w \in \delta(i)\}$

• $\mathtt{E}\ (\mathtt{IsA}) \subseteq \mathtt{D} \times \mathtt{W}$. If `p` and `q` are two classes in a `Telos` knowledge base and class `q` is a specialization of class `p` for a time interval `i` and `r` is the proposition defining this relationship, then $\forall w \in \delta(i)\ < \gamma(\mathtt{r}), w > \in \mathtt{E}(\mathtt{IsA})$.

[20] $i_1 \sqsubseteq i_2$ means that interval i_1 is contained in interval i_2.

• If q is an instance of p for a time interval j, then C(p) contains a tuple $< \gamma(\mathtt{q}), w >$ for all $w \in \delta(j)$.

•E(InstanceOf), contains a tuple for any proposition defining an instantiation relationship for all possible worlds in the associated time interval.

A.3.1 Assertional Component

Satisfiability of formulas in a possible-worlds model M is defined in terms of the relation $\models$, a binary entailment relation between model-worlds pairs and formulae.[21]

•$(\mathtt{M}, w) \models$ $\mathtt{prop(p,x,y,z,t)}$ iff $\forall w \in \delta(t) < \gamma(p), \gamma(x), \gamma(y), \gamma(z), \gamma(t), w > \in \mathtt{V(prop)}$ and $\exists \mathtt{q} \in \mathtt{KB}$ such that $from(\mathtt{p}) = x,\ label(\mathtt{p}) = y,\ to(\mathtt{p}) = z, when(\mathtt{p}) = t$ and $\gamma(\mathtt{q}) = \gamma(\mathtt{p})$.

•$(\mathtt{M}, w) \models$ $\mathtt{instanceOf}(x, y, t1, t2)$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(instanceOf)}$ and $\forall w' \in \delta(t1)\ < \gamma(x), w' > \in \mathtt{C}(y)$ and $\exists \mathtt{p} \in \mathtt{KB}$ such that $from(\mathtt{p}) = \mathtt{q} =< x, instanceOf, y, t1 >, label(\mathtt{p}) = instanceOf$, $to(\mathtt{p}) = \mathtt{InstanceOf}, when(\mathtt{p}) = t2$ and $< \gamma(\mathtt{p}), w' > \in \mathtt{E(InstanceOf)}$, $< \gamma(\mathtt{p}), t2 > \in \mathtt{B}$.

•$(\mathtt{M}, w) \models$ $\mathtt{isA(x,y,t1,t2)}$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(isA)}$, and $\exists \mathtt{p} \in \mathtt{KB}$ such that $\mathtt{p} =< \mathtt{x, isA, y, t1} >$ and $\forall w \in \delta(t1)$, $< \gamma(\mathtt{p}), w > \in \mathtt{E(IsA)}$ and $\exists \mathtt{q} \in \mathtt{KB}$ such that $from(\mathtt{q}) = \mathtt{r} =< \mathtt{x, instanceOf, IsA, t1} >, label(\mathtt{q}) = instanceOf$, $to(\mathtt{q}) = \mathtt{InstanceOf}), when(\mathtt{q}) = \mathtt{t2}$ and $\forall w \in \delta(t2)\ < \gamma(\mathtt{r}), w > \in \mathtt{E(InstanceOf)}$ and $< \gamma(\mathtt{q}, t2 > \in \mathtt{B}$.

•$(\mathtt{M}, w) \models$ $\mathtt{att}(x, y, t1, t2)$ iff $\forall w \in \delta(t1) < \gamma(x), \gamma(y), \gamma(t1), \gamma(t2), w > \in \mathtt{V(att)}$ and $\exists \mathtt{q, r, s, Q} \in \mathtt{KB}$ such that $\mathtt{q} =< \mathtt{x, att, y, t1} >, \mathtt{r} =< \mathtt{q}, instanceOf, \mathtt{Q}, t3 >$, $\mathtt{s} =< \mathtt{r}, instanceOf, \mathtt{InstanceOf}, t2 >$, $\mathtt{Q} =< \mathtt{CL}, \ldots, \mathtt{Class}, \ldots >$ and $\forall w \in \delta(t3)$ $< \gamma(\mathtt{r}, w > \in \mathtt{C(Q)}$ and $\forall w \in \delta(t2) < \gamma(\mathtt{s}, w > \in \mathtt{E(InstanceOf)}$ and $< \gamma(\mathtt{r}), t2 > \in \mathtt{B}$ where CL is a class of which x is an instance for a time interval containing $t1$.

• $(\mathtt{M}, w) \models$ $(t1 = t2)$ iff their corresponding individuals are both in Q(w) or both in D and they are named by the same standard name.

• If x, y are time intervals with $x = (x_1, x_2)$, $y = (y_1, y_2)$, then $(\mathtt{M}, w) \models$ $x\ \theta\ y$ iff $(\mathtt{M}, w) \models \tau(\theta, x, y)$, where $\tau(\theta, x, y)$ is a point-based expression.

• $(\mathtt{M}, w) \models \alpha$ $\mathtt{and}\beta$ iff $(\mathtt{M}, w) \models \alpha$ and $(\mathtt{M}, w) \models \beta$

• $(\mathtt{M}, w) \models \alpha$ $\mathtt{or}\beta$ iff $(\mathtt{M}, w) \models \alpha$ or $(\mathtt{M}, w) \models \beta$

• $(\mathtt{M}, w) \models \alpha\ \Rightarrow \beta$ iff $not((\mathtt{M}, w) \models \alpha)$ or $(\mathtt{M}, w) \models \beta$

• $(\mathtt{M}, w) \models \alpha\ \Leftrightarrow \beta$ iff $(\mathtt{M}, w) \models \alpha \Rightarrow \beta$ and $(\mathtt{M}, w) \models \beta \Rightarrow \alpha$

• $(\mathtt{M}, w) \models$ $\mathtt{not}\ \alpha$ iff $not((\mathtt{M}, w) \models \alpha)$

For well-formed formulas that involve quantifiers, we distinguish two cases: one where the variable bound by the quantifier are non-temporal, and one where the variable ranges over a

[21]The model-world pair is just a shorthand for (M,KB,B,w). As it will be clear in the sequence, satisfiability depends on the model M of the knowledge base KB, the knowledge base itself, the belief time relation B and a possible world w. For brevity we will use only the pair $(\mathtt{M}, w)$ but will mean the quadruple (M,KB,B,w).

set of time intervals. In the first case we have for a variable assignment i mapping variables in elements of $\mathtt{S(C)}$:

• $(\mathtt{M}, w) \models (Forall\ x/C)\ \alpha$ iff for every variable assignment i' mapping variables in elements of $\mathtt{S(C)}$ and differing from i possibly on the value of $x, (\mathtt{M}, w) \models \alpha$

• $(\mathtt{M}, w) \models (Exists\ x/C)\ \alpha$ iff for some variable assignment i' mapping variables in elements of $\mathtt{S}(C)$ and differing from i possibly on the value of $x, (\mathtt{M}, w) \models \alpha$

In the second case :

• $(\mathtt{M}, w) \models (Forall\ t/\mathtt{TimeInterval})\ \alpha$ iff $\forall w' \in \mathtt{W}$ such that wRw' or $w'Rw$, $(\mathtt{M}, w') \models \alpha$

• $(\mathtt{M}, w) \models (Exists\ t/\mathtt{TimeInterval})\ \alpha$ iff $\exists w' \in \mathtt{W}$ such that wRw' or $w'Rw$,
and $(\mathtt{M}, w') \models \alpha$

where `TimeInterval` is the built-in class comprising all time intervals.