Correlating Documents with Common Knowledge by Using Attribute Links

Friso Schutte
Institute of Computer Science
Foundation of Research and Technology - Hellas

July 1996

Contents

List of Figures

Preface

This report is a result from research done at the Institute of Computer Science at the Foundation of Research and Technology - Hellas, Heraklion, Crete, Greece.

I would like to thank Prof. Panos Constantopoulos for allowing me to do this Erasmus project on the beautiful island of Crete. Thanks goes of course also to Dr. Martin Dörr, with whom I had long and interesting philosophical discussions, and to all other people from the information systems group who helped me in getting things straight.

Raxit and Gaelle were very nice company for the delicious daily lunches. Many thanks goes to Joanna, who was so patient all the time. Back we go to the rainy streets of Amsterdam!

Chapter 1

Introduction

In conventional information retrieval systems a keyword-based text representation approach is used. The underlying assumption is that keywords, which are either assigned to these texts or occur naturally in titles and abstracts, are sufficient for representing the topics in the text. *Indexing* is defined as the process of converting a collection of documents into a form suitable for easy search and retrieval. An *indexing language* is the language used to describe documents and requests, i.e. 'a controlled set of terms selected from natural language and used to represent, in summary form, the subjects of documents' (ISO2788 [ISO86]). The elements of the indexing language are *indexing terms*. Normally, the indexing terms are topical subjects (unambiguous keywords) and a document is represented as a list of keywords[1]. A *thesaurus* is the vocabulary of an indexing language (i.e. all possible indexing terms), formally organized so that the *a priori* relationships between concepts (e.g. "broader" and "narrower") are made explicit. Thus, a thesaurus may help with creating or extracting knowledge from specific lists of keywords. But a list of keywords is still a list and this means that specific relationships between terms within a list can not be specified. A list of keywords may be faceted in different facets, but even this does not remove certain ambiguities. In a faceted system a specific relation between one term in a facet and one term in another facet that holds for one document can not be specified. Unless it is obvious and unique from the combination of the keywords, a faceted list of keywords may be ambiguous. For example, different facets in a system for classification of scientific papers could be 'subject', 'method applied', and 'result'. But for the same subject one method may yield a different result than a second method.

Soergel [Soe95] pointed out some problems with a specific existing thesaurus, the Art and Architecture Thesaurus (AAT). But implicitly it also showed some weaknesses of thesauri in general, as stated in ISO2788. Moreover, thesauri may be regarded as a knowledge base problem in it's own right [Dör96]. Knowledge bases and thesauri represent a kind of domain knowledge and as such can be used for intelligent intermediary systems, i.e. systems which help the users defining or refining their topics of interest. A thesaurus captures synonyms, and associative and hierarchical relations. A knowledge base could go beyond this, by capturing more semantical relations and complex structures. Some systems, such as EP-X, make an explicit distinction between the knowledge base, which describes domain knowledge, and the data base, which indicates which documents are relevant to particular topics represented in the knowledge base [SSGC89]. In our research the distinction is not that clear, since we try to incorporate the documents (or more precisely, the surrogates of the documents) into the knowledge base, just as other concepts within the data model.

In this paper we investigate the use of semantic networks for indexing documents. Semantic networks are a well established mean for representing knowledge with great expressive

[1] In this context an indexing term usually is a noun *or* a noun phrase.

power. Although adding AI techniques to the knowledge base, such as domain expertise, search expertise, and user models such as semantic networks, has been successfully incorporated in many (intelligent) information retrieval systems, in most systems however, the governing framework is still the keyword approach: that is, texts are represented and searched using keywords.

Some research suggest an emphasis on the text-based conceptual roles of keywords using text grammars [RS93], while other pay special attention to the aggregated structure of documents, and especially the mixture of the contents of a document and factual knowledge about the document [RF96].

In this paper however, we investigate the semantic relationships between keywords and emphasize on these relationships using an explicit representation of relationships between a document and 'domain knowledge'. The *Semantic Index System (SIS)* is an object-oriented semantic network knowledge base management system, which allows such structures. Different conceptual models for correlating objects with common knowledge will be presented.

Chapter 2 will give some background, necessary for understanding the SIS data model and the underlying knowledge representation language Telos, which are explained in Chapter 3. In Chapter 4 a description of an example application of the SIS will be given and some knowledge representation problems are described. In the end some conclusions about the preceding chapters will be drawn.

Chapter 2

Some Background

This chapter describes the necessary background for the Semantic Index System and the knowledge representation language Telos. In Section 2.2 data models in general will be discussed. This will lead to an explanation of semantic data models and object-oriented data models. There has been a tremendous amount of research in all kinds of data models. Usually a distinction is made between traditional record-based models and modern models, which incorporate a much higher level of abstraction. For more extensive overviews of different kinds of data models see e.g. [BM88] or [PT88].

The next section will discuss semantic networks. Although semantic networks can be traced back as far as last century (Peirce's existential graphs), renewed interest came when their use in computer science became obvious.

But first a somewhat philosophical discussion of some of the notions used in databases, artificial intelligence, and information retrieval will be presented.

2.1 Data, Knowledge, and Information

Research in computer science is divided in many areas. Moreover, other sciences, such as information science or library science, are somehow related. When reading articles, the terminology used depends a lot on the writer's background. The term "knowledge" traditionally is used in artificial intelligence. Although intelligence is not a well defined notion, it for sure has something to do with knowledge (such as common sense or expert knowledge). Knowledge-based systems, like expert systems, usually have a base with rules and an inference mechanism for handling the rules. "Data", as opposed to knowledge, merely involves simple facts, such as "John works at IBM". But research in databases led to the conclusion that adding knowledge to the base, such as "John is a Person" and "All Persons have an Age", would improve retrieval effectiveness of data. In this way we could define a knowledge base as an extension of a database. Sometimes the term "information base" is used to refer to both databases and knowledge bases. For a deeper discussion about the combination of AI and databases, see [Bro88].

Meanwhile "information" was the main term used in information retrieval [Rij79]. One could say that traditionally information retrieval refers to the retrieval of relevant documents from a collection of documents, by means of a query. The effectiveness of an IR system is usually expressed in terms of *precision* (the proportion of retrieved documents which are relevant), and *recall* (the proportion of relevant documents retrieved). The main problems are caused by the facts that one concept might be expressed by different words (synonyms), and one word or expression might refer to different concepts (homonyms).

The difference with databases lies in the difficulty of representing documents, such that the content can be searched semantically, and in the nature of the problem, which is by definition incomplete and imprecise. Data retrieval usually deals with exact matching of queries in an artificial query language. In information retrieval the goal is to find the most relevant documents by means of queries, preferably in natural language. Direct integration of AI in IR systems has been most clearly in the use of expert systems as intelligent intermediary systems. Expert systems traditionally have a rule-based representation of expert knowledge. The main advantages of production rules is their logical foundation and that they are easy to implement with languages as Prolog. The main disadvantage is their slowness in processing. In contrast with rules, semantic networks have a clear graphical form, which may be of help in the knowledge acquisition phase.

For an article on the integration of AI and IR, see [Bro87]. Although somewhat sceptic about the use of expert systems in IR, she emphasizes that developments in IR can benefit from AI, especially at the conceptual level. Obviously, IR should benefit from AI research done in areas like knowledge representation (representing a document such that the system "knows" some things about the document contents) and natural language processing (like in the processing of queries).

Although documents originally refer to books, articles or reports, with the popularity of multimedia, today the term often also refers to images or sounds. The effectiveness of an information retrieval system, or a data base system, lies for a great deal in knowledge representation. What we are aiming at are "intelligent information systems", a term which captures all previous mentioned terms.

2.2 Data Models

A *data model* can be seen as 'a collection of mathematically well-defined concepts that help one to reason about data and to express the static and dynamic properties and integrity constraints for an application' [BM88]. As such, a data model defines the formal structure of an information system. The traditional data models include network, hierarchical, and relational models. These are also called *record-based*. They generally promote efficient storage and retrieval of fixed information structures. But their usefulness is restricted by their failure to take into account the semantics of the database [Ken79]. This led to a rise of the so-called *semantic data models* [HK87, PM88, SLHS93]. Some research suggested extensions of existing models (e.g. [Cod79, Sto84]), but nearly all were influenced by the artificial intelligence field of knowledge representation. The semantic data models emphasize on the user's perception of data rather than its physical representation. Most of them are strongly influenced by semantic networks. In fact, semantic data models could be seen as semantic networks with a certain emphasize on the separation of schema and data, that is abstracting the structure of data from the data itself.

A semantic model represents a piece of the "real world", generally referred to as an *enterprise*. The important elements within an enterprise, the *objects* or *entities*, and the structural relationships between these elements are represented in the so-called *semantic schema*. An instance of a schema typically contains objects which conceptually correspond directly to physical or conceptual objects in the world.

A very related generation of data models are the *object-oriented data models* [BM91, ZM89]. The object-oriented paradigm became popular in all the fields of computer science, and thus it also effected research in databases. But the difference between the two models is not mentioned in all literature, since semantic data models are generally object-oriented in the sense that they provide the typical relationships between objects (concepts, entities) like aggregation, classification, and generalization. *Aggregation* refers to the relationship between an object and its attributes. E.g. a person can be seen as an aggregation of its legs, arms, head, etc. In another way, a person can be seen as a social object and thus as an aggregation of the set of properties such as speaks, works, has_friends, etc. *Classification* is a mechanism for representing the instance-of relation. As such it can be compared with the element operator in set theory. John can be seen as an instance of the class person with elements being individual persons. When a class has as its elements classes, then it is called a metaclass. *Generalization* has the subset operator in set theory as its analog. Classes are specialized along so called isa hierarchies. The class woman, with elements being all individual women, is a specialization of the class person, with elements being all individual persons. Of course it all depends on the person modeling an enterprise and on the enterprise being modeled, since a person may also be seen as a metaclass with elements being the two classes woman and man. In that case the class woman is an instance of person. The main reason why classification and generalization should not be uniform to aggregation (i.e. why the instance-of and isa relation is different than normal attributes), is the support for inheritance of properties. A network should show a hierarchy, such that domain-specific and general properties are not being mixed.

The difference between semantic and object-oriented models seems to be that the first attempts to provide *structural abstractions* and the latter *behavioral abstractions*. Object-oriented models define operations locally within an abstract datatype. Semantic data models are close to knowledge representation languages and object-oriented models to programming languages. But the two models seem to have grown towards each other and together they provide ways of expressing the structural and behavioral aspects of complex objects (see e.g. [BR84, PMD95, Tar92] for the integration of semantic and object-oriented ideas).

2.3 Semantic Networks

While semantic data models arose from the field of databases, semantic networks became popular in AI. The semantic schema should be seen as a kind of semantic network. Databases are intended for a large amount of data and efficient handling of queries. Most data models therefore specify both a notation for describing the data (data definition language) and operations for manipulating the data (data manipulation language). AI on the other hand is concerned with sophisticated manners for representing the data. The processing of a semantic network is usually not included in the definition of a semantic network, but of course a semantic network is only useful when inference can be done. The data structure of nodes and a set of inference procedures (such as inheritance) together are called a *semantic network representation scheme*. Thus the distinction between a semantic data model and a semantic network representation scheme is not as clear as we would like. To make things worse, both sometimes lack procedural part (or manipulation language). So let us just call the whole thing (e.g. the SIS data model): *"object oriented semantic network data model"* [CD93].

A semantic network is a conceptual structure for the representation of knowledge as a pattern of interconnected nodes and arcs. Both the nodes and the arcs may be labeled. A node represents a real-world concept or entity. Arcs represent relationships between concepts. Other knowledge representation paradigms are e.g. the frame theory and different kinds of logic such as first order logic. Knowledge represented in one can usually also be represented in another.

Before the wide spread use of computers, graph representation mechanisms evolved from research in logic, psychology and linguistics, in the beginning of the 20th century. However, the first implementations of semantic networks were developed in the late 1950s and early 1960s. These were designed as intermediate languages for machine translation systems. Still they are very much used in the field of natural language processing. But different networks

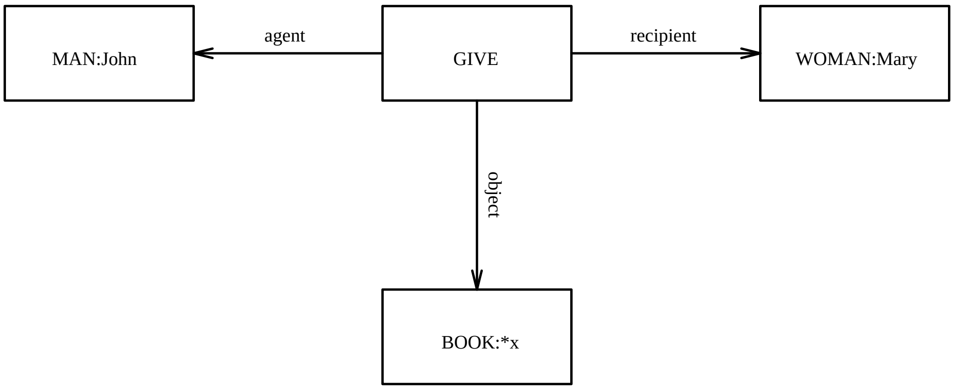

Figure 2.1: Graph with variables for "John gives Mary a book"

arouse all over the world and different types of links and concepts were often not recognized as such. Semantic networks seemed to be constructed somewhat intuitively. During the 1970s some (now famous) theoretical critiques on the formal background of semantic networks where made [Bra79, Woo75], and during the 1980s the boundary between semantic networks and different forms of logic and representation formalisms such as frames, blurred. The expressive power is not the main issue anymore since knowledge can generally be represented in both network and linear notations. But semantic networks remain an easy to read, a kind of natural, and efficient way of representing knowledge.

The simplest networks used in AI are so called *relational graphs*. These are graphs in which each node represents a concept and each arc between two nodes represents a specific relationship between the corresponding concepts. Simple natural language sentences can be mapped easily into a network.

One way for representing the sentence "John gives Mary a book" is given in Figure 2.1, which is adopted from Sowa's conceptual graphs[1] [Sow84]. Note the usage of variables.

When certain nodes represent entire propositions, a network is called a *propositional network*. For this too, different notations are in use. One way is to put a box around a proposition and use it just as any other node (e.g. [Sow84]). Another way appears in the case structure representation. The proposition in Figure 2.1 would be represented by one node with links to 'John', 'book', and 'Mary', as well as a 'verb' relationship link to 'give'. In SNePS [SR92] for example, a sentence like "Sue thinks that Bob believes that a dog is eating a bone" is represented like in Figure 2.2 (example from [Sow87]).

Classical semantic networks are purely declarative, but extensions can be made to allow for procedural attachment. Procedures as well as properties could be inherited through the type hierarchy.

[1] Sowa defines a conceptual graph as a graph representing just one proposition, whereas a semantic network is much bigger and also represents generalization, classification, assocational relations or whatever. A conceptual graph thus can be mapped to a semantic network, and is only meaningful when mapped.

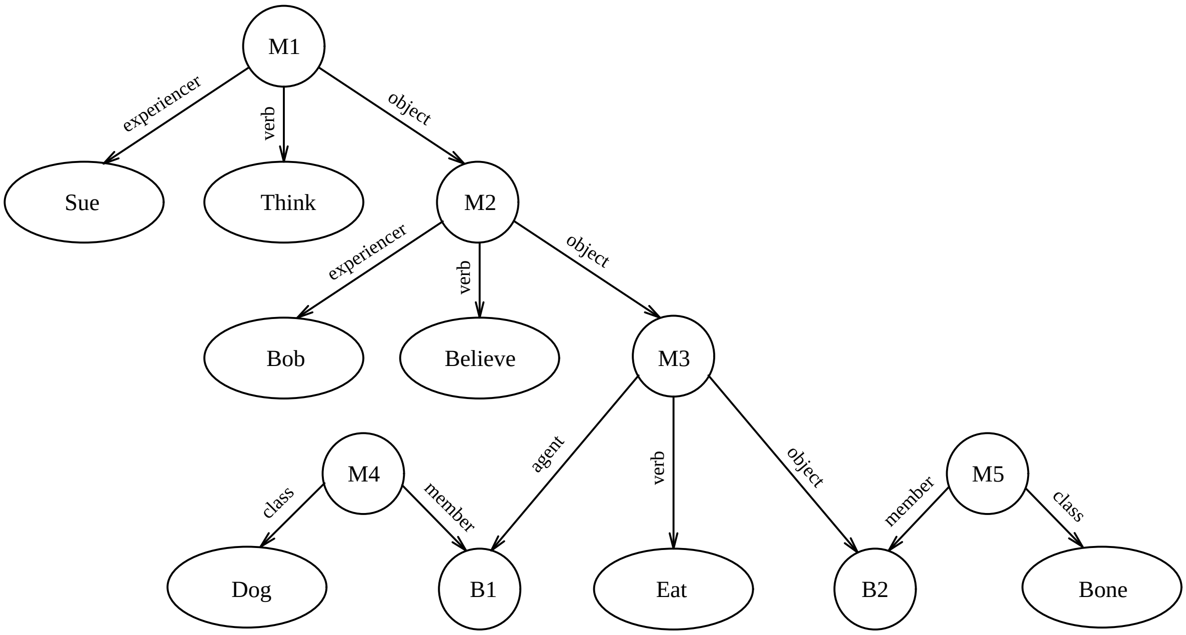

Figure 2.2: SNePS representation of “Sue thinks that Bob believes that a dog is eating a bone”

Chapter 3

The Semantic Index System and Telos

The Semantic Index System (SIS) [CD93] is a tool for describing and documenting large evolving varieties of highly interrelated data, concepts, and complex relationships, as opposed to large homogeneous populations in fixed formats, handled by traditional DBMS. It consists of a storage mechanism and a generic user interface. For the purpose of this paper, we will focus on the knowledge representation side of the SIS, leaving the implementation details for what they are.

Knowledge is represented using the Telos knowledge representation language [MBJK90]. Therefore, the SIS data model is based on the Telos data model. The next section will give an overview of Telos. After this the main features of the SIS will be given.

Section 3.3 will describe how a knowledge base system as the SIS can be used for information retrieval.

3.1 Telos

Telos has been presented as a language for representing knowledge about information systems. It is strongly influenced by work done in semantic networks, semantic data models, and object-oriented representations. This section describes the most important features of Telos, which could be divided into a structural, temporal, and assertional component. However, since the SIS only uses the structural component, this component will get most of the attention.

The standard reference for Telos is Mylopoulos et al. [MBJK90]. For a more detailed description of the design and features of Telos, see [KMSB89]. For the theory behind it, a formal (possible worlds) semantics is given in [Ple93].

3.1.1 Telos knowledge bases

A Telos knowledge base is a collection of *propositions*. A *proposition* is the single representational unit. They are divided into two disjoint categories, *individuals* and *attributes*. *Individuals* represent entities, and *attributes* represent binary relationships between entities or other relationships. One of the most important features of Telos is that individuals and attributes are treated uniformly. This is why an attribute can also represent a relationship between relationships. Propositions are organized along the three dimensions, mentioned in Section 2.2, aggregation, classification, and generalization.

A Telos attribute proposition consists of a source, label, destination, and time-stamp, which can be retrieved through the functions `from(p)`, `label(p)`, `to(p)`, and `when(p)`. An attribute, therefore, may be represented as a 4-tuple. However, the representational capacities of temporal knowledge were not implemented in the SIS. For the purposes of this paper, it is thus sufficient to represent an attribute as a 3-tuple, e.g. `[Martin, age, 35]`. A structured object may then be represented as an aggregation by a collection of attributes that have a common proposition as the source, e.g. {`Martin, [Martin, age, 35], [Martin, homeAddr, '21 Elm Avenue']`}.

The classification dimension specifies that every proposition is an instance of at least one class. Classes are themselves propositions, and therefore instances of other, more abstract classes. Propositions having no instances are called *tokens*, propositions having only tokens as instances are called *simple classes*, propositions having only simple classes as instances are called *metaclasses*, propositions having only metaclasses as instances are called *metametaclasses*, etc. This results in a potentially infinite instantiation hierarchy. Classes having instances along more than one such plane are called *ω-classes*. One example of an ω-class is the class `proposition`, having all propositions as instances.

Propositions can also be specialized or generalized along ISA hierarchies. This can only be the case if the proposition is a class and not a token. The class `Person` for example, may have subclasses `Researcher`, `Professor`, `Student`, etc. A class can have more than one superclass and therefore multiple inheritance of attributes is supported. In the case where two superclasses have the same name for an attribute with different meaning, the subclass must specify explicitly from which superclass it inherits which attribute.

From the preceding examples it should be clear that a Telos knowledge base can be represented as a semantic network, although a full semantic network may look somewhat complex. Figure 3.1 shows a simplified portion of a knowledge base, in which the mentioned examples are depicted in graph form. Since all nodes and edges are propositions, every node and edge should be connected with the class `proposition` by an instance relation. In Figure 3.1 this is simplified for readability purposes. Also some other ω-classes as `attribute` or `individual` or omitted for simplicity.

3.1.2 Assertion language

Telos also includes an assertion language, which resembles first order logics. Influenced by deductive databases, the assertion language is used in two classes of assertions: *integrity constraints* and *deductive rules*. The former constrain the knowledge in the knowledge base, while the latter are used to derive implicit knowledge. In addition, it can be used to pose queries. The SIS is designed as a very fast system. The assertion language is one of the features from Telos which are not implemented in the SIS to avoid efficiency drawbacks in large databases.

3.1.3 Updating the knowledge base

Propositions can be added to the knowledge base through the `TELL` command, and can be updated with the `UNTELL` or `RETELL` command. For querying the knowledge base, Telos offers `RETRIEVE`, which uses only temporal and structural information to answer queries, and `ASK`, which uses all the knowledge available. The inference mechanism of the query language has not been implemented in the SIS due to efficiency drawbacks. The only commands maintained are `TELL` and `RETELL`. The syntax of these commands are almost the same as in the original design of Telos. In the next section they will be described more thoroughly.

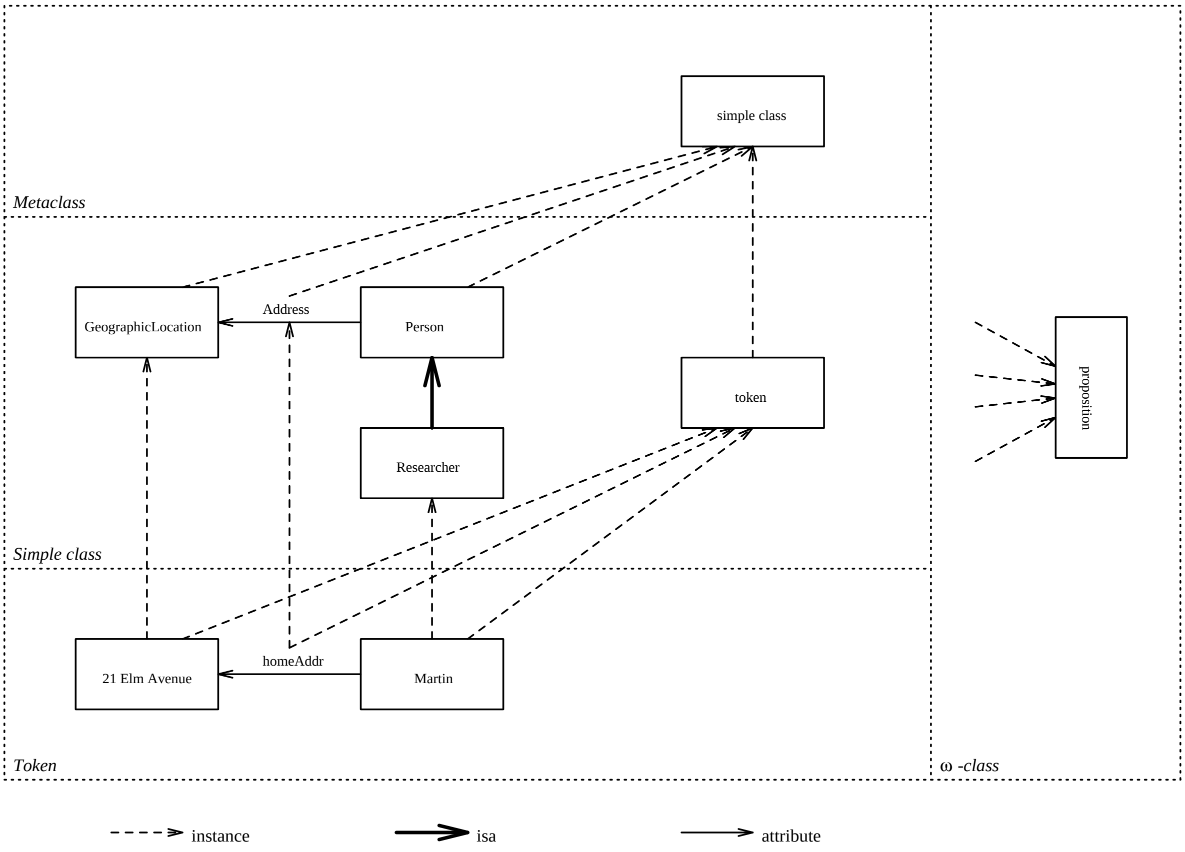

Figure 3.1: Simplified part of an example Telos knowledge base in graph form

3.2 SIS

The SIS is developed at the Institute of Computer Science, Foundation of Research and Technology (FORTH), Heraklion, Greece, in the framework of the European ITHACA ESPRIT project concerning software repository systems. The SIS consists of a persistent storage mechanism based on an object-oriented semantic network data model, and a generic interactive user interface to insert and retrieve information in various ways. It is implemented in C++.

Various applications have made use of the SIS as the kernel, mainly in the fields of software static analysis, class management and software reuse (the Software Information Base (SIB) [CJMV95, CD95b]), and cultural documentation (the CLIO Museum Information System [Con94, CCD92]). Ongoing research is dealing with more possible applications, such as thesaurus management, mechanical fault documentation, and indexing of annotated images [CD95a].

An overview of the Semantic Index System can be found in [CD93]. The Telos implementation, called the SIS data entry language, is described in [DKT95].

3.2.1 The SIS data model

The SIS uses the Telos language for its representational framework. Actually, only the structural part of Telos was kept. All "AI features" were removed. This means that the assertion language, inference mechanism, and temporal component, all three mentioned in the previous section, were not implemented.

The assertion language is not needed from the database point of view. The SIS is not an expert system, and therefore does no reasoning. It merely states the facts given by the designer. This means that the designer should be precise in adding data, but also that the system will not crash if some data turns out to be impossible in the real world (e.g. that John is 184 years old).

The inference mechanism from Telos as designed in the original language is not supported. Instead, a complete database query language based on navigational set operators is implemented in the SIS. Following links from object to object, it is possible to retrieve any kind of data wanted.

The temporal component is not implemented because it was not necessary in the framework of the first application, namely the Software Information Base. This was a system intended for the reuse of software objects [CJMV95].

Classification, specialization, and aggregation are supported. Since classification states that every proposition must be an instance of a class, the SIS offers an initial population. A SIS knowledge base consists of a number of built-in classes, which can not be changed by the user. The number of meta levels is supported until the metametametametaclass (`M4_Class`), which in practice seems to be sufficient for useful applications. A Telos proposition is called an *object* in SIS. Every user-defined object is an instance of a built-in class, even if the object is also an instance of a user-defined class. The built-in classes form both an instantion and an ISA hierarchy. They are depicted as a kind of Venn diagram in Figure 3.2.

As an example, consider the class `Individual_M1_Class`, which is a subclass (specialization) of `M1_Class` and `Individual`, `M1_Class` is a subclass of `Telos_Class`, and `Telos_Class` is a subclass of `Telos_Object`.

Three built-in individual classes exist for primitive values: `Telos_Integer`, `Telos_Real`, and `Telos_String`.

Objects in a SIS knowledge base should be given a unique identifier (within their scope), which is as close as possible to the natural language description of the corresponding entities, concepts, relations or notions in the real world.

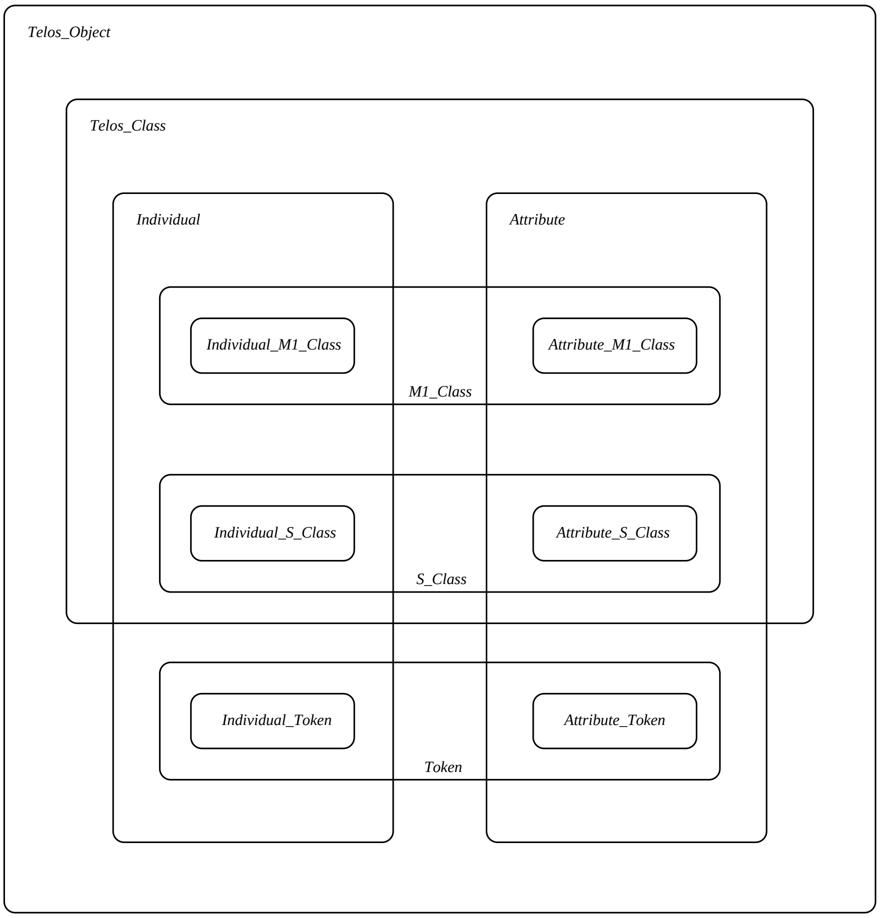

Figure 3.2: Built-in classes in SIS Telos

An object can be added to the knowledge base using the TELL or RETELL statement. One session of TELL or RETELL statements is called a transaction. A transaction should begin with BEGINTRANSACTION and end with ENDTRANSACTION. The following transaction shows how the propositions from Figure 3.1 could be programmed.

```
BEGINTRANSACTION

TELL Individual GeographicLocation in S_Class
end GeographicLocation

TELL Individual Person in S_Class with
    attribute
      Address: GeographicLocation
end Person

TELL Individual Researcher in S_Class isA Person
end Researcher

TELL Individual (21 Elm Avenue) in Token, GeographicLocation
end (21 Elm Avenue)

TELL Individual Martin in Token, Researcher with
    Address
      homeAddr: (21 Elm Avenue)
end Martin

ENDTRANSACTION
```

Instead of implementing attributes implicitly, as in the individuals Person and Martin in the example above, it is also possible to implement attributes explicitly. In this way it is not only possible to state that an attribute is an instance or subclass of another attribute, it is also possible to make an attribute link to an individual. If we want to state, for example, that the fact that Martin is living at 21 Elm Avenue is supported by his driving licence, we could make a link between the attribute homeAddr and a new individual (Martin's Driving Licence). Below is the SIS Telos code for this example. Figure 3.3 shows it in graph form.

```
TELL Attribute homeAddr
    from: Martin
    to:   (21 Elm Avenue)
  in Token, Address with
    attribute
      Supported_by: (Martin's Driving Licence)
end homeAddr
```

Note the difference between the representations in Figure 3.3 and Figure 2.2. Section 2.3 mentioned different methods for representing propositional networks. The term proposition in

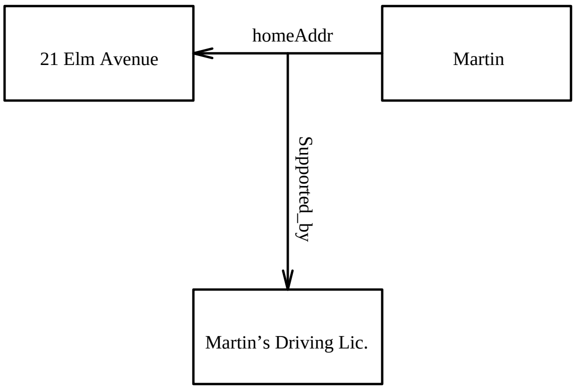

Figure 3.3: Example of an attribute having an attribute

	destination	
source	attribute	individual
attribute	no	yes
individual	no	yes

Table 3.1: Possible attribute links

semantic networks is used somewhat different than in Telos. A proposition in this way should be seen as a set of Telos propositions or SIS objects. One way for representing a sentence like "Martin's driving licence supports that Martin is living at 21 Elm Avenue" would be to make one proposition of the "proposition" "Martin is living at 21 Elm Avenue" by drawing a box around it. Another possibility was to make a "molecular-node" which is linked with all the concepts in the proposition. This is sometimes called the case structure notation.

One disadvantage of these methods is that you have to name the node which contains the other concepts. Most often this results in machine generated names. This is against the idea of SIS and Telos, namely that every node has a name which is as close as possible to the natural language term of the concept. The SIS offers extensive browsing possibilities which would suffer largely from names which are not immediately clear to the user. Telos offers the possibility of generating the same result without extra nodes by making a relation between attribute and an individual.

Although both attributes and individuals are first-class citizens of the population, they are not completely interchangeable. An attribute can not be a subclass or instance of an individual. This would not make much sense. But an attribute does have to be an instance of at least one attribute class. An attribute class is also called a *category*. It is also not possible (at least not in the current version) to make an attribute link from an individual to an attribute. An attribute can not be the destination of another attribute. This means that also an attribute link from an attribute to an attribute is not yet supported (Table 3.1).

When specifying a link from an individual to an attribute, the model should make an inverse link from the attribute to the individual. The direction of an attribute is a linguistic one, since an inverse should always be possible to create (e.g. `Supported_by` instead of `Supports`).

Deleting or updating an object can be done through the `RETELL` statement, which has nearly the same syntax as `TELL` (see [DKT95]).

3.3 Using the SIS for Information Retrieval

Data in the SIS can be used as a big knowledge base which can be queried and from which different kinds of information can be retrieved. We can use it as a traditional document retrieval system by applying the following steps:

1. Define a simple class `Document` as a generic structure (case, frame) with the usual attributes, as author, title, abstract, keywords, etc.

2. Make every document you wish to index an instance of `Document` and assign the appropriate attribute values.

All the attributes could have `Telos_String` as destination. The system could then perform keyword searches on e.g. the list of keywords assigned to each document, or it could do full text retrieval on the abstract. But a system like this makes very little use of the capabilities of the SIS. Besides, it would probably be slower than commercially available IR systems which only serve this purpose.

A first and obvious step in making the system more useful and which makes use of an important feature of the SIS, would be to specialize the class `Document` into subclasses, such as `Book, Article`, etc.

Less obvious, but important when it comes to searching for books of a specific author, is to cluster all authors into individual authors. An authority file or some other kind of standardization could be of help here. Every different author is an object in its own right. Thus the author of a document is not a string but an object. The authors would be instances of a new class `Author`, with its own attributes, such as nationality, date of birth, etc., some of which could be inherited from a more generic class `Person`. This requires the author attribute from `Document` to point to an individual of type `Author`. Likewise other attributes as editor, publisher, etc. should be clustered.

While this may seem all very logical, the question of representing the contents of a document remains one of the most difficult topics.

3.3.1 Beyond word-based systems

In an article in a special issue on the integration of AI and IR of *Information Processing & Management*, Croft states [Cro87]:

> Representation of documents with complex, controlled languages that incorporate phrases, thesaurus, and other structural relationships has never been shown to be effective in terms of retrieval performance. (...) Reducing words to stems and the incorporation of a simple thesaurus with synonym relations is the only vocabulary control that has been shown to have definite advantages.

This was in 1987. Since then a lot of research has been made in the design of controlled vocabularies for use in IR systems [Sve89], in incorporating structural relationships and usage of thesauri as a search aid [Kri93], in sophisticated methods for automatic text classification using natural language processing or information extraction techniques [RL94], in the use of thematic roles or text grammars for the representation of the contents of a document [WD91, RS93], and in the use of all kinds of knowledge bases in the form of rules, frames, or semantic networks in order to improve search in an IR system [SSGC89].

What all these techniques have in common, is that they are semantic approaches as opposed to the traditional, statistical word-based approaches which have been used in IR since many years. Some limitations of word-based techniques deal with synonyms and homonyms, but also with the context words appear in. Some words are only good indexing terms in specific phrases or in local contexts. The word "passed" and "away" are not generally associated with dying, but the phrase "passed away" is. This indicates that in some cases automatic indexing can be difficult.

Some techniques

The SIS is suited for applications in which many cross-references are needed. In the case of document retrieval, we could represent the content as a complex structure. This needs to be done manually and can be very time consuming, but may be of use in determining different concepts and the relationships between concepts. Such ambiguities as caused by homonyms would be eliminated.

We should also keep in mind such bases where documents are images [CD95a]. With the content of an image we then refer to different annotated objects and their interrelationships within the image, and not, as is usual in literature about content analysis of images, to the colors, shapes etc. The latter would be difficult to represent in a system like the SIS, since it generally involves mathematics with real numbers which can not be represented by a discrete, finite set of relations.

As a matter of document or annotated image indexing, we should make a distinction between two cases. The first is the case where the elements are more or less disjoint. In this case a document is regarded as information in its own right. The main indexing mechanism is classification. An example would be the indexing or classification of software artifacts. The second is the case where the elements have large overlaps, which would be the case in very specialized domains. In this case documents (or images) where the same objects appear in all documents (or images), but different relationships between the objects hold, are being indexed. Suppose you want to index all Dutch landscape paintings. In much of these paintings there appears a windmill, a river, and a cow. Sometimes the cow is left to the windmill and sometimes right, or behind, etc. Note that this example is not a real example of common data, unless the pictures show *the same* windmill, cow, and river. Another example is the case where the domain knowledge is a story and the history of the story has to be represented also. Or the data of an encyclopedia is indexed semantically and the sources where the data came from has to be included. In this case the sources have to be included in the knowledge base, just like other knowledge, by linking these sources to the encyclopedical knowledge.

The ideal would be to have one semantic network for the complete domain and links from every document to specific nodes or edges of this network. The two cases are illustrated in Figure 3.4.

Users are often interested in the relationships among concepts rather than in single concepts. Most systems therefore allow search with multiple keywords or keyword phrases, with the assumption that the relation between the keywords follows from the keywords.

EP-X (Environmental Pollution eXpert [SSGC89]) is a special domain intelligent IR system. This system interprets a keyword list, and even detects ambiguous keyword phrases (although it is not clear if it also detects ambiguous relations between keywords). EP-X has domain knowledge stored in the form of hierarchically defined semantic primitives and frames. It captures both synonyms and homonyms, as well as broader and narrower terms. Documents are stored in a database as frame instantiations. In order to support more efficient processing of specific queries, concepts in the domain knowledge are linked with relevant documents (see Figure 3.5).

Thematic roles [WD91] or text grammars [RS93] are used in representations for qualifying keywords or phrases with their thematic or conceptual roles in a text. Such representations are potentially more powerful than pure keyword representations used in conventional IR

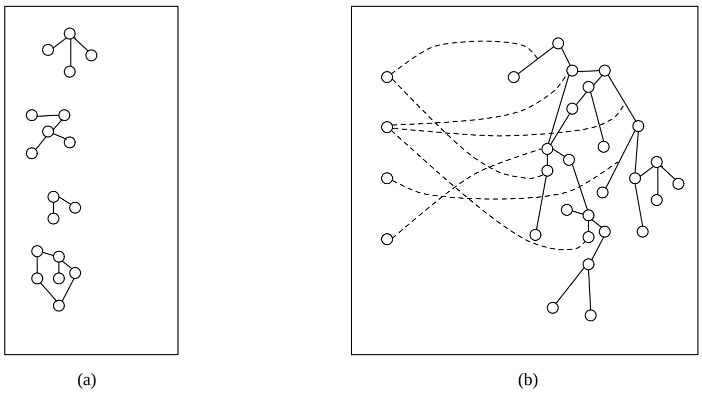

Figure 3.4: Two cases of document indexing. (a) Documents are represented by structures which have little to do with each other. (b) Documents are represented by links to a domain knowledge base.

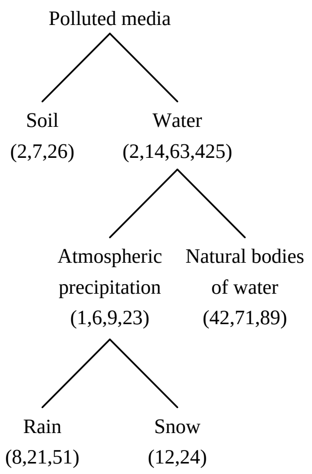

Figure 3.5: Partial hierarchy from EP-X showing document identifiers attached to relevant nodes. Taken from [SSGC89].

systems. In contrast to the semantics, the thematic or conceptual role of a keyword or phrase can not be determined a priori, but depends on the context. This suggests a different kind of network, or multiple networks and hierarchies in the domain knowledge.

Information extraction [RL94] is yet another way with which people attacked the problems statistical word-based IR systems deal with. Automatic high precision text classification seems only possible if a system could understand a document. But full natural language processing is impractical and currently not able to generate in-depth analyses of unconstrained text. Information extraction is only concerned with the domain-specific portions of the text. Portions that are not relevant to the domain are ignored. In doing this an information extraction system needs a specific domain knowledge base.

Natural language processing techniques such as information extraction help in determining if a document is relevant. It can be used for automatic text classification a priori, or in determining if a text is relevant to a query in a full text database.

3.3.2 A SIS application

Although it would be interesting to use the SIS as a document retrieval system taking all the previous in account, we depart in our application a little from the traditional field of document retrieval. The application involves the ULAN, which is a structured collection of artist and architect data (it will be described in more detail in the next chapter). As such it can and should be seen as a controlled vocabulary. But by incorporating the sources of the data, and the sources of the sources, it reflects some ideas comparable with those in IR systems.

Although controlled vocabularies, thesauri, or authority files should serve as intermediary systems which help the user "to get what it wants", implementing them in a form suitable for usage requires that they are not seen as passive dictionaries, but as active, evolving components in a bigger network [Dör96]. The application of the SIS to the ULAN reflects certain ideas about the indexing problem in the SIS. Key principles in this are the metamodeling and rich reference capabilities.

Chapter 4

Linking Documents to Common Knowledge by Means of an Example

This chapter describes different approaches to the conceptual modeling of a large controlled vocabulary, or encyclopedical data. The ULAN database is a production of the Getty Art History Program. The data from the ULAN will be used to illustrate the general case of correlating documents to common knowledge. Some advantages of using an object oriented system instead of a relational database will be shown. Specifically, the SIS offers techniques which are of interest for storing the ULAN data. In fact, some problems which occur in modeling the ULAN data can be generalized to problems which occur in modeling of every kind of encyclopedic data. Some conceptual models will be discussed.

4.1 The Getty Art History Information Program

The J. Paul Getty Trust is a private operating foundation that seeks to make a significant contribution to the vitality of the visual arts in the areas of conservation, scholarship, and education.

The *Getty Art History Program (AHIP)* is one of the six operating programs of the Trust[1]. The AHIP works to enrich the content of the computer networks of the future by encouraging those concerned with preserving the cultural heritage to collaborate in building a cultural information infrastructure. The idea is to have access to cultural heritage all over the world. The need for common standards for electronic art information is the underlying theme.

The main advantages of standards are:

- improved information retrieval
- protection of the long-term value of data
- sharing of information from heterogeneous sources.

AHIP works in collaboration with institutions in America and overseas to evolve and establish crucial standards. Three structured vocabulary resources for use in creating and retrieving information about works of art have been produced so far: the *Art & Architecture Thesaurus (AAT)*, the *Union List of Artist Names (ULAN)*, and the *Thesaurus of Geographic Names (TGN)*. All three are accessible through specialized software.

[1] The others are the J. Paul Getty Museum, the Getty Center for the History of Art and Humanities, the Getty Conservation Institute, the Getty Center for Education in the Arts, and the Getty Grant Program.

4.2 The Union List of Artist Names[2]

The *Union List of Artist Names (ULAN)* is a huge, structured collection of personal data from artists and architects, mainly from America and Western Europe. In the following the term artist refers to both artists and architects.

Every artist in the ULAN has associated with it a list of (variant) names, biographic data, and bibliographic data, as well as the projects, which contributed the data. An artist and its associated data is called a *cluster*. Contributors of the data range from abstracting and indexing projects working with primary documents to object-based collections[3], as well as the department of AHIP responsible for the coordination of controlled vocabulary resources, the Vocabulary Coordination Group (VCG). The latter contributed only a limited number of records, mostly it added names for disambiguating artists.

The most important features of the ULAN are:

- Whereas traditional authority files enforce usage by clearly indicating a single "preferred" form of name, the ULAN presents neutral views of its constituent authorities. A cluster has an entry form name, zero or more project-preferred names (name forms that are preferred by the various contributing projects, but that do not match the ULAN entry form), and non-preferred variant names (contributed variant names that do not match the ULAN entry form). Nevertheless, a cluster *has* an entry form name, which is the preferred name of the preferred contributing project. Thus a preferred name is tacitly implied.

- The clusters are unique in their size and completeness. The ULAN contains some 200,000 names representing approximately 100,000 individual artists. Every cluster includes biographic data, such as the artist's nationality, the artistic role(s), and life dates.

- Bibliographic citations provide for source information. Citations are linked by contributing projects to an artist, a specific name, or an element of the biographic information. The sources are clustered in much the same way as the names, in the sense that citations, identified differently by the contributing projects, but referring to the same source were clustered in a single bibliographic cluster. The ULAN database contains more than 190,000 references to nearly 24,000 sources.

A schematic representation of the ULAN is given in Figure 4.1.

When building the ULAN, the AHIP had different users in mind. Among others, the ULAN can be used as a tool for:

- building local authority files

- validating names and some other data

- choosing which form of an artist's name should be used for cataloging purposes or in essays

- research or as a reference tool.

[2]Most of the information in this section is extracted from [ULA94].

[3]For the interested reader, the nine contributing projects are: Avery Index to Architectural Periodicals, Bibliography of the History of Art/Bibliographie d'Histoire de l'Art, Census of Antique Art and Architecture Known to the Renaissance, Foundation for Documents of Architecture, Getty Center for the History of Art and the Humanities - Photo Study Collection, The J. Paul Getty Museum, The Provenance Index, Witt Checklist of Painters c1200–1976, and Witt Computer Index.

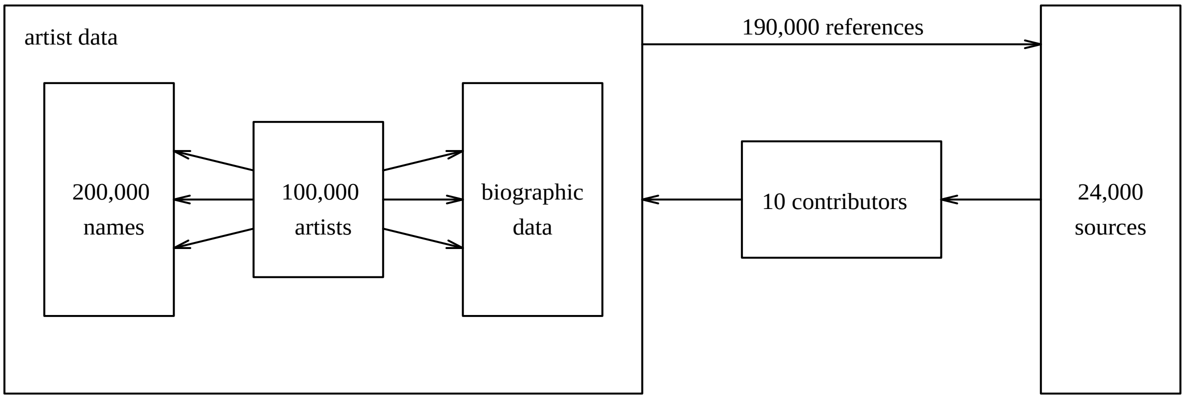

Figure 4.1: Schematic representation of the ULAN

The ULAN is available in electronic form as ULAN:ART. ART stands for Authority Reference Tool and is an AHIP software program for generic use with controlled vocabularies. ULAN:ART represents only a part of the whole database. ULAN:ART is designed as a user friendly tool, which allows rapid access to name, biographic, and bibliographic data. Additionally, it has a simple ("cut-and-paste") function for copying terms into word processing documents or database records.

It does not, however, support searching on data other than the name of the artist, neither does it support "broad" searches where more than 200 clusters match a query. Also, the bibliographic data is simplified in the sense that a cumulative, alphabetized list of bibliographic references is linked to a single cluster, not to specific names or biographic data.

The complete ULAN database is not intended for the broad audience, but mainly as an editing environment for the VCG itself. The database contains much more information than ULAN:ART. The biographic data e.g. is divided in smaller pieces such as the role (architect, painter) and even relations to other artists, such as family relations or teacher-student relations, are included. It is imaginable that data about the artists will be extended in the future, such that even more information about the life of the artist, the works of the artist, the family of the artist, etc. will be included.

4.3 The SIS and ULAN

The SIS research group at the Institute of Computer Science, FORTH, has built a prototype with the same data as the ULAN. This SIS system is intended as an alternative for the current relational database. As such it would be suited as an editing environment. The object oriented representations, rich reference possibilities, and graphical user interface should play a key role.

A proposed conceptual model, called the Vocabulary Control System (VCS) schema, is described and motivated in [DC96][4]. The VCS schema is the logical equivalent of the three applications of the AHIP, the AAT, ULAN, and TGN. The notions of the three systems are integrated into one model. In this paper, however, we will only be concerned with the ULAN data. To give an idea of what and how knowledge about an artist is represented, Figure 4.2 gives an example, copied as is, from the current SIS prototype. It shows the artist Vermeer, with all "known" characteristics. The value '21th century' is just a standard value without any meaning. There are two arrows pointing from 'born' and from 'died' because that was contributed by two different projects. The two projects apparently agreed on the birth and

[4] In the meanwhile some changes have been made in the actual implementation.

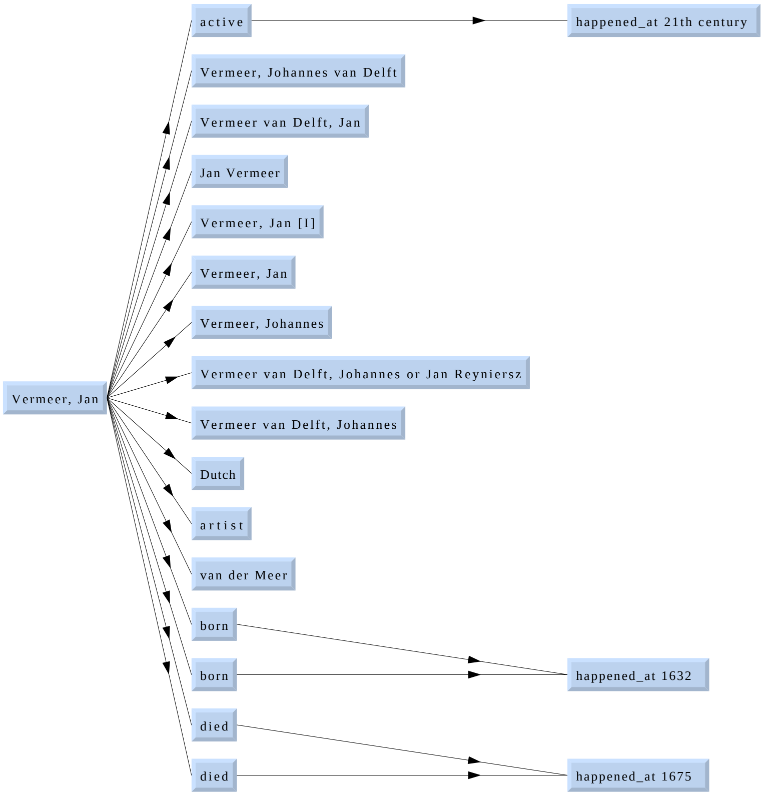

Figure 4.2: Example from SIS prototype. Person Characteristics Tree of Vermeer.

death date.

In contrast with ULAN:ART it is possible to search for a pattern in everything that is recorded about an artist, and it is possible to show different types of graphs according to the user's needs. Figure 4.2 for example shows the characteristics of Vermeer (Person Characteristics Tree), whereas Figure 4.3 graphically shows the submission of data about Vermeer by the Provenance Index (PersonSubmission Star View). The combination of the submitter and the person forms a key, which is taken as the main node. From the graph we can see that the Provenance Index contributed more than one non-preferred name and one preferred name (Vermeer, Jan) for the person 'Vermeer, Jan'. Thus the ULAN entry form name matches the Provenance preferred name.

However, the bibliographic data is not implemented. In the next sections the data model is explained in more detail, and proposals for alternatives, as well as for an extension, allowing for bibliographic data, are made.

4.3.1 The prototype model

The SIS prototype model is depicted in Figure 4.4. The instances and some of the isa-relations are not shown in the figure for clearity. All the objects are instances of `S_Class`. Furthermore, most of the individuals are subclasses of `ThesaurusConcept`, which is an instance of `ThesaurusNotionType`.

One interesting thing is the original idea that the attributes `name`, `role`, `as_person_name`, `as_member_in`, `as_person_role` are all subclasses of `timed_relation`, which is a relation between two individuals with an attribute `valid_during`. For example, an artist may have been called x till a specific date and after this date he or she may have been called y. The

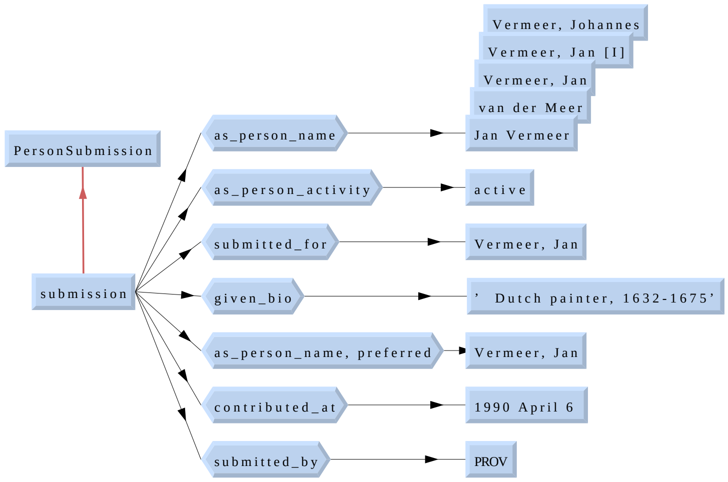

Figure 4.3: Example from SIS prototype. Star View of the submission of data about Vermeer by the Provenance Index.

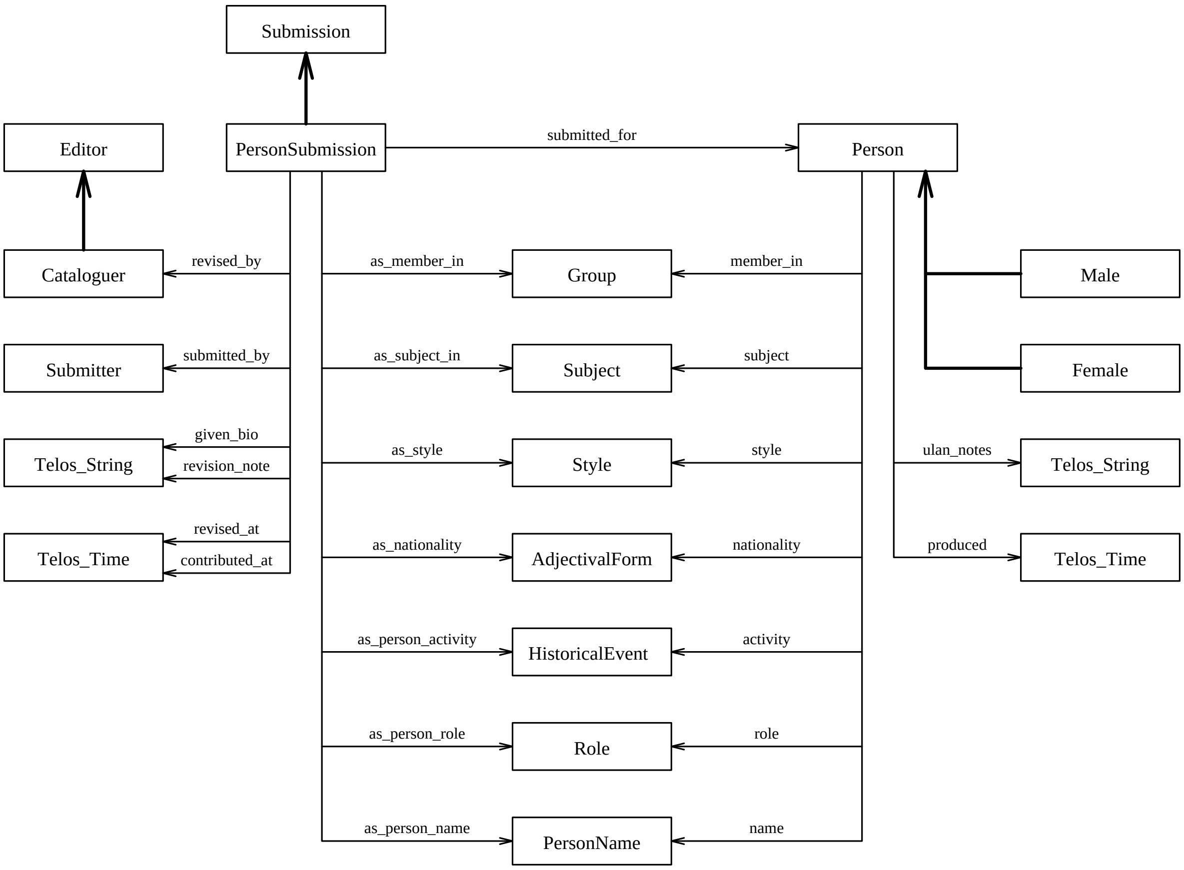

Figure 4.4: Part of the ULAN model as used in the SIS application. Instance relations are not depicted. Isa relations are depicted by fat arrows.

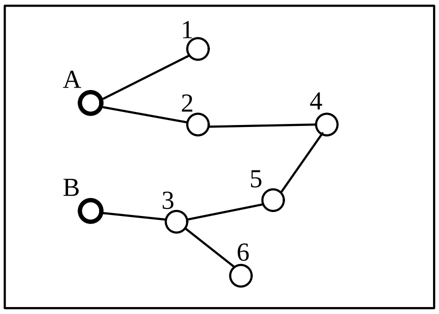

Figure 4.5: Two pieces of data, A and B, and references to sources, 1 to 6

timed_relation attribute is not used in the current version.

Another interesting idea was the attribute found_in, which was linked from Individual to Source. This means that every individual could have a source responsible for its presence. This is again not implemented in the actual system. Although at first sight it may seem sophisticated, it is more a kind of hack, probably due to time pressure. First, not every individual can be found in a source, only a part of it. Second, the relation between the source and the contributor which linked a specific source to a specific artist, name, or element of the biographic information gets lost.

The good thing about this construction is that it shows the *dynamic* capabilities of the SIS. Since Source is also an individual, we can express dynamically sources of sources. In a relational database this would be arranged as a *static* process with an x number of stages. A piece of data has a source which in turn has a source.

4.3.2 Some problems in art information

In the art world there are a number of possible problems possible in determining "who is who" and "who made what" or "what is made by who".

Some problems derive from the fact that different artists share the same name (synonym) and the same artist is known under different names (homonym). For example, an artist may never have been existed. What were thought to be two persons, actually turned out to be one and the same. An additional problem in art history is that many art works are not signed by the artist (e.g. medieval art). A particular piece of art may be completely different from all other pieces of the same period. Does this imply a new previously unknown artist? Or if it resembles other work, how can we be sure if it is from the same artist of which comparable work is found? Or was it made by a student of his? Or a teacher?

References to literature or other references may reveal a lot of the answers. A dynamic source-citation mechanism would deserve preference over a static mechanism. Some data can be traced back six stages where other only one. And the further back the more reliable the source (not always!). And how should we store all these data in a database?

Consider Figure 4.5. The existence of person A is confirmed by two sources, 1 and 2, of which source 2 has a reference to 4. Person B is confirmed by source 3 which in turn has references to 5 and 6. Source 5 citated 4, but in doing so, it misspelled the name, creating a non existing person. So it looks like person A and B are actually the same person. And what about source 6? Perhaps the citation of 5 to 4 was incorrect, or perhaps 6 made an error? This is only to show some problems which *could* arise when tracing sources.

Consider only an artist like "the Master of the Paradise Garden", who is only known as such because the actual name of the painter of the Paradise Garden is not known, and the painting resembles none of the works from that time.

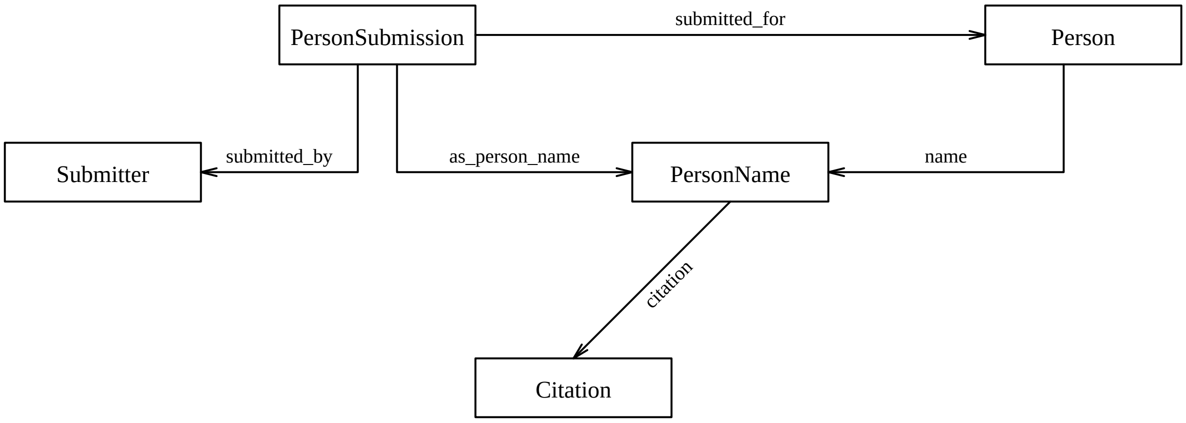

Figure 4.6: Linking the citation to the individual.

An artist like Barthel Schongauer, who is also known as B.G. Monogrammist, is said to be controversial in the sense that his existence is being doubted.

The art historian Panofsky from this century wrote several transactions about art history. In order to illustrate a specific movement or style, he is said to have sometimes *invented* artists who typically fitted in the movement or could serve as a prototypical artist of a certain style.

Consider also the role of database systems. The clustering of people is only the first step. Citations have also been clustered. But the data in the ULAN database is not yet as reliable or consistent as we would like it to be. Some problems arise because of the merge of different databases, all with their own formats. Some drawbacks were encountered because different people added the data. And sometimes a database was just not intended for search on non-conventional fields. For example, when a search on all Dutch artists in the base is performed, the user would not only have to keep in mind the different names for the Netherlands which it has and which it had during the passage of time, it would also have to know the different formats a country name has been inputted in the database. Furthermore, the nationality of an artist may have different degrees of doubt. How do we represent this? For the example of Dutch artists, I encountered the following name forms: Dutch [?], DUTCH, Dutch, Dutch[?], early Netherlandish, early Netherlandish?, Flanders, Flemish, Flemish[?], Netherlandish, Netherlandish[?], Netherlands or German, Netherlands, Netherlands[?], North Netherlandish, probably Flemish. Of course, we can search on partial strings with boolean operators, such as 'neth* and du*' but this does not seem satisfying.

The clustering of data is in fact the same as making objects in an object-oriented system. A system like the SIS encourages the usage of standards for the input of data. Search on every kind of data can easily performed, even if nobody is interested (at the moment) in retrieving specific kind of data.

4.3.3 Incorporating bibliographic information

First attempt

A first attempt to include bibliographic information in the model of Figure 4.4, will be to add a class `Citation`, which we define as a combination of a source and additional information such as the volume, pagenumber, etc., called 'collation' in the original database. All the data which require a source (the role, name, life dates, etc. of an artist), can get an additional attribute `citation` (or `found_in`). Figure 4.6 shows the result.

The problem with this approach is that the information about which project used which source might get lost in some cases. The simple case where this goes wrong is illustrated in Figure 4.7. We would like to somehow put parentheses around specific combinations of

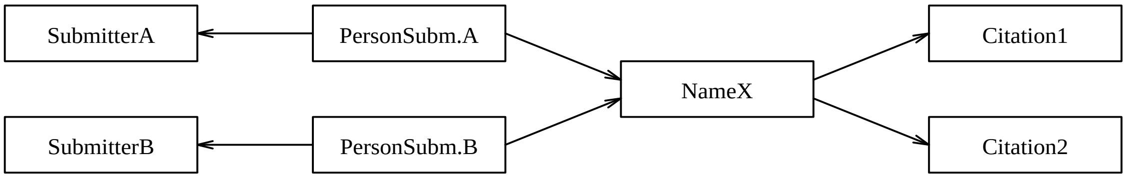

Figure 4.7: Which citation belongs to which submission, and thus to which submitter?

submissions and citations. The generalization of this problem is well known in the field of knowledge representation.

To circumvent the loss of information we could create a combination node of `Citation` and `Submitter` (called something like `Submitter+Citation`), which has a link to both the citation and the submitter.

A better way

What exactly is the citation used for? The answer to this is that the citation is used by a contributing project to justify its contribution. Thus it is not a bad idea to make a link from the attribute `as_person_name` (or `as_person_role` etc.) to the citation. In this way it would preserve the information of which piece of data can be found in the citation, as well as which project contributed the citation (implicitly by the PersonSubmission node). Figure 4.8 illustrates the situation. A query like "which contributing projects used the Thieme-Becker as a source" could be efficiently answered by following links from Thieme-Becker back to the actual contributors.

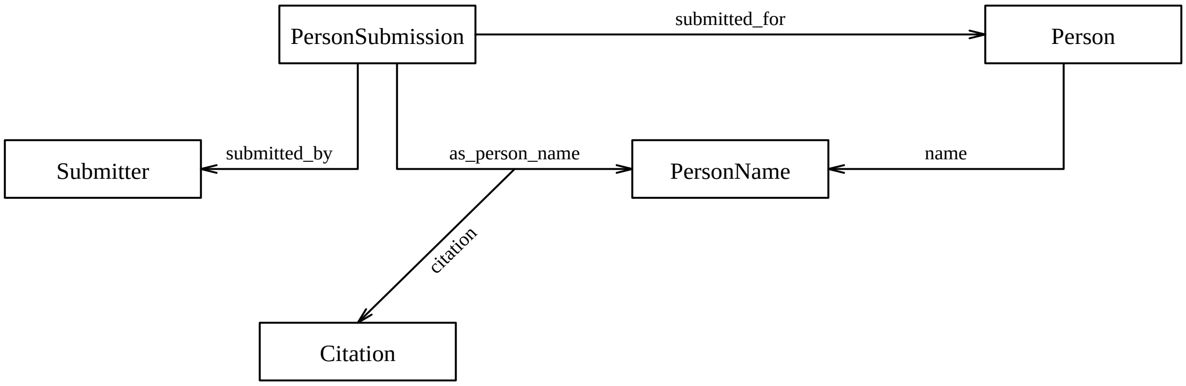

Figure 4.8: Linking the citation to the attribute.

As for representing this knowledge in Telos, it should be noted that the `as_person_name` attribute should be declared explicitly, since it has an attribute itself (Section 3.1). When declaring an attribute implicitly it is possible to give it no external label, which means that the attribute will get a system generated label. When declaring attributes explicitly this is not (yet?) possible. This means that the instances of `as_person_name` should get a label (such as the combination of the PersonSubmission and the Person). This poses no real problem, since the input of data does not have to occur by writing Telos transactions. Special purpose entry forms, which translate the input to Telos code, can easily be constructed.

More models

Now that we are talking about attributes on attributes, we could wonder if a change in the original model (Figure 4.4) is wishful. The individual `PersonSubmission` is an aggregation of a submitter (contributing project) and specific data it submitted for a specific person. As such it has its own id. This sounds like it is designed especially for the complicated tables of the relational database. In a relational database it is helpful to give a table a key in the form of an id.

In the SIS version we could make directly (analog to above) a link from the attribute `name` (from `Person` to `PersonName`) to the submitter of this data (Figure 4.9).

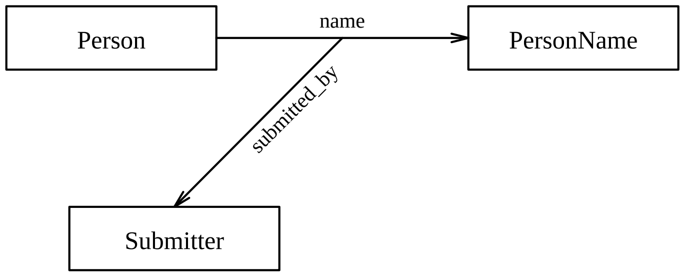

Figure 4.9: Linking the submitter to the attribute.

An interpretation of this situation is "ProjectA stated that PersonB has NameC as a name". To incorporate bibliographic information it would be nice to draw another link from `claimed_by` to `Citation`. We can not make only a link to the `name` attribute, because then we loose the information between which Submitter added which Citation. We can also not link the Citation to the Submitter, because then we loose the information between which Citation made the Submitter add which piece of data. Of course we can make two links, one to `name` and one to `Submitter`. But this seems to include more links than necessary.

A solution would be to have `Citation` linked to `claimed_by`. The link would express something like "confirmed by". A link from `name` to `Citation` would implicitly be derived. We should make a metamodel where we make a distinction between reality relations and claimed relations. The former would group relations such as `name`, `role`, etc. The latter would group the claimed by, or confirmed by relations. Both metaclasses (categories) would have an attribute `claimed_by` which would point to, say, an individual.

In this way we have provided for the dynamic needs explained earlier. We can also change the Citation node into a Submitter node and vice versa. The point is that it is symmetrical, but with subtle difference in interpretation. If we let the Submitter point to the name relation and the Citation to the relation between the Submitter and the name relation, then we are expressing *the submitters opinion on what the truth was.* If we let the Citation point to the name relation and the Submitter to the relation between the Citation and the name relation, then we are expressing *the submitters opinion on what the citation says is the truth.* In both ways we can extend the model to include the opinion on the opinion on the opinion etc. of Mr. X. Figure 4.10 illustrates the case.

Mapping a relational database

Instead of a model like above, it is of course also possible to leave the model as it is in Figure 4.4. To allow for bibliographic references, the link to the 'as_x' relations (`as_person_name`, `as_person_role`, etc.) can be made, and even sources of sources could be expressed just like in Figure 4.10.

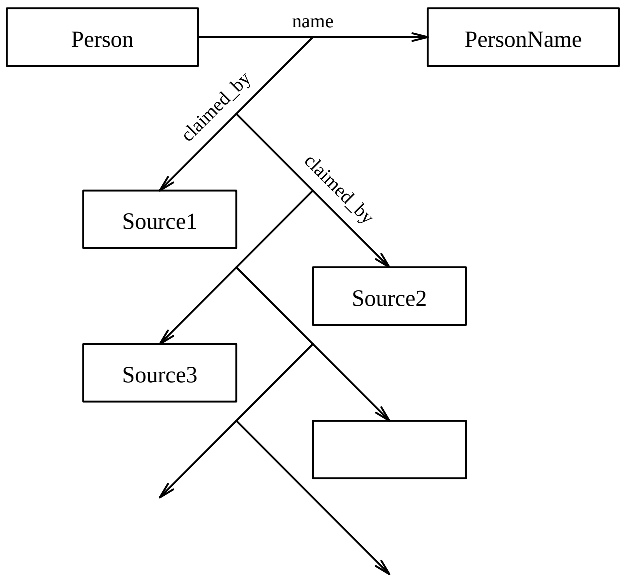

Figure 4.10: Citations of citations.

Instead of the model of dynamic relations on relations, it is also possible to make only one relation on a relation, just like in Figure 4.8. To allow for the dynamic needs we define a link from `Citation` to `Citation`. We then can get something like in Figure 4.11. This is probably

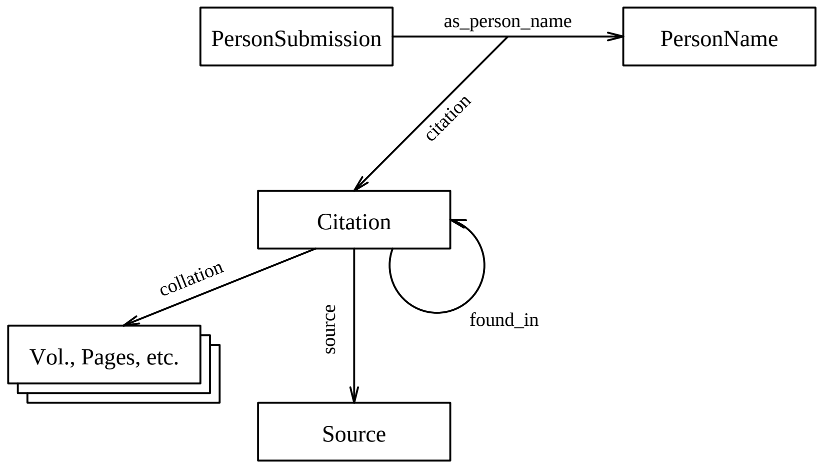

Figure 4.11: Simple recursion of citations.

the simplest way to make a model for the ULAN which includes bibliographic references.

It seems like some key nodes (e.g. `PersonSubmission`) are especially designed for relational tables. A typical relational database has a table for different entities. Each table has its own fields. The combination of specific fields normally form a key, but a table typically has one key entry. This key is usually a unique number. In order to link two tables, the key is used to refer to the record (entity). For example, a citation may have associated with it a list of biography table keys. This forms a link from a citation to a specific biography of a specific artist.

A more entity oriented system would be suited better for this kind of data. The keys would turn into object identifiers. In the SIS datamodel, which is a kind of extended entity-relationship model, an entity key distinct from a natural language identifier is not necessary.

Different truth values

A metamodel which includes 'reality' and 'claimed' links can be used to denote different degrees of truth about relations between objects (statements). Instead of a source claiming that person x is a painter, it may also be the case that the source explicitly claims that x is not a painter. Or a second source which says that what the first source said about x is false. Contradicting sources may be discovered quickly, but may also be hidden in different granularities.

A source A may state that El Greco is born in a house in Fodhele in the Heraklion district on Crete in Greece. A second source, B, may even deny that El Greco was born on Crete.

Chapter 5

Conclusion

In this paper the possibilities of attribute relationships in an object oriented, semantic datamodel are discussed. Some background about datamodels in general, and object oriented and semantic datamodels in particular, as well as background about knowledge representation in general, and semantic networks in particular, is surveyed. The fields of databases, information retrieval, and artificial intelligence are being integrated in order to provide sollutions for knowledge representation problems which are found in all three disciplines.

The Semantic Index System is a knowledge base management system well suited for research in semantic models for complicated real world problems with a large amount of interrelated entities. The system can be used both as a normal object oriented database system and as an intelligent information retrieval system based on a semantic representation of the data.

A specific existing prototype application of the SIS to data from the Union List of Artist Names (ULAN) is analyzed at the conceptual level. The ULAN is a structured list of data about artists and architects. Different names from different sources for the same artist are clustered together with some biographical data. It is meant as an authoritative system. From an object oriented perspective, different conceptual models are discussed for the representation of not only the simple data itself, but especially the inclusion of different sources of the data. The necessity and feasibility of sources of sources are discussed. The emphasis is on the use of attribute links on attributes, which is allowed in the SIS.

With the example of the ULAN data we discussed only a specialization of the more general problem of including sources to data in all kinds of systems, such as controlled vocabularies or encyclopedical systems. The research is not at all finished. This paper only tries to move the focus on objects to relationships between objects.

Bibliography

[BL85] Ronald J. Brachman and Hector L. Levesque, editors. *Readings in Knowledge Representation.* Morgan Kaufmann, Los Altos, CA, 1985.

[BM88] Michael L. Brodie and Frank Manola. Database management: A survey. In J.W. Schmidt and C. Thanos, editors, *Fundamentals of Knowledge Base Management Systems.* Springer-Verlag, 1988. Also in [MB88].

[BM91] Elisa Bertino and Lorenzo Martino. Object-oriented database management systems: Concepts and issues. *IEEE Computer*, 24(4):33–47, April 1991.

[BMS84] Michael L. Brodie, John Mylopoulos, and Joachim W. Schmidt, editors. *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages.* Springer-Verlag, 1984.

[BR84] Michael L. Brodie and Dzenan Ridjanovic. On the design and specification of database transactions. In Brodie et al. [BMS84].

[Bra79] Ronald J. Brachman. On the epistemological status of semantic networks. In N.V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, New York, 1979. Also in [BL85].

[Bro87] H.M. Brooks. Expert systems and intelligent information retrieval. *Information Processing & Management, Special issue: Artificial Intelligence and Information Retrieval*, 23(4):367–382, 1987.

[Bro88] Michael L. Brodie. Future intelligent information systems: AI and database technologies working together. In Mylopoulos and Brodie [MB88].

[CCD92] Maria Christoforaki, Panos Constantopoulos, and Martin Dörr. CLIO: An object-oriented model of cultural data. Technical Report ICS-FORTH.MUIS.92.1, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, November 1992.

[CD93] Panos Constantopoulos and Martin Dörr. The Semantic Index System: A brief presentation. Working Paper #6, Information Systems and Software Technology Group, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, August 1993.

[CD95a] Panos Constantopoulos and Martin Dörr. An approach to indexing annotated images. In David Bearman, editor, *Multimedia Computing and Museums, Selected Papers from the Third International Conference on Hypermedia and Interactivity in Museums, San Diego, CA, USA, Oct '95*, pages 278–298, Pittsburgh, 1995.

[CD95b] Panos Constantopoulos and Martin Dörr. Component classification in the Software Information Base. In O. Nierstrasz and D. Tsichritzis, editors, *Object-Oriented Software Composition*. Prentice-Hall, 1995.

[CJMV95] Panos Constantopoulos, Matthias Jarke, John Mylopoulos, and Yannis Vassiliou. The Software Information Base: A server for reuse. *VLDB Journal*, 4(1):1–43, 1995.

[Cod79] E.F. Codd. Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems*, 4(4):397–434, 1979. Also in [Sto88].

[Con94] Panos Constantopoulos. Cultural documentation: The CLIO system. Technical Report FORTH-ICS-TR-115, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, January 1994.

[Cro87] W. Bruce Croft. Approaches to intelligent information retrieval. *Information Processing & Management, Special issue: Artificial Intelligence and Information Retrieval*, 23(4):249–254, 1987.

[DC96] Martin Dörr and Maria Christoforaki. The Getty AHIP VCS Schema. Technical Report VCS.FORTH.96.1, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, March 1996.

[DKT95] Martin Dörr, Polyvios Klimathianakis, and Manos Theodorakis. *SIS - Data Entry Language User's Manual, SIB Static Analyser and Class Management Runtime System, Version 1.3.* Institute of Computer Science, FORTH, Heraklion, Crete, Greece, November 1995.

[Dör96] Martin Dörr. Authority services in global information spaces, a requirements analysis and feasibility study. Technical Report ICS-FORTH-TR-163, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, February 1996.

[HK87] Richard Hull and Roger King. Semantic database modeling: Survey, applications, and research issues. *ACM Computing Surveys*, 19(3):201–260, 1987.

[ISO86] ISO2788. Documentation — guidelines for the establishment and development of monolingual thesauri. Second Edition, International Organization for Standardization, 1986.

[Ken79] William Kent. Limitations of record-based information models. *ACM Transactions on Database Systems*, 4(1):107–131, 1979. Also in [MB88].

[KMSB89] Manolis Koubarakis, John Mylopoulos, Martin Stanley, and Alex Borgida. Telos: Features and formalization. Technical Report FORTH-ICS-TR-018, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, February 1989. Also as Technical Report KRR-TR-89-4, Computer Science Department, University of Toronto.

[Kri93] Jaana Kristensen. Expanding end-users' query statements for free text searching with a search-aid thesaurus. *Information Processing & Management*, 29(6):733–744, 1993.

[Leh92] F. Lehmann, editor. *Semantic Networks in Artificial Intelligence*. Pergamon Press, Oxford, 1992.

[MB88] John Mylopoulos and Michael L. Brodie, editors. *Readings in Artificial Intelligence and Databases.* Morgan Kaufmann, San Mateo, CA, 1988.

[MBJK90] John Mylopoulos, Alex Borgida, Matthias Jarke, and Manolis Koubarakis. Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4):325–362, 1990.

[Ple93] Dimitris Plexousakis. Semantical and ontological considerations in Telos: a language for knowledge representation. *Computational Intelligence*, 9(1), February 1993.

[PM88] Joan Peckham and Fred Maryanski. Semantic data models. *ACM Computing Surveys*, 20(3):153–189, 1988.

[PMD95] Joan Peckham, Bonnie MacKellar, and Michael Doherty. Data model for extensible support of explicit relationships in design databases. *VLDB Journal*, 4(2):157–191, 1995.

[PT88] Walter D. Potter and Robert P. Trueblood. Traditional, semantic, and hypersemantic approaches to data modeling. *IEEE Computer*, 21(6):53–63, June 1988.

[RF96] Thomas Rölleke and Norbert Fuhr. Retrieval of complex objects using a four-valued logic. In *SIGIR Proceedings of the 19th Conference on Research and Development in Information Retrieval, Zürich, Switzerland.* ACM, 1996. To appear.

[Rij79] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 2nd edition, 1979.

[RL94] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, 1994.

[RS93] D.V. Rama and Padmini Srinivasan. An investigation of content representation using text grammars. *ACM Transactions on Information Systems*, 11(1):51–75, 1993.

[SE87] Stuart C. Shapiro and David Eckroth, editors. *Encyclopedia of Artificial Intelligence.* John Wiley & Sons, Inc., 1987.

[SLHS93] John L. Schnase, John J. Leggett, David L. Hicks, and Ron L. Szabo. Semantic data modeling of hypermedia associations. *ACM Transactions on Information Systems*, 11(1):27–50, 1993.

[Soe95] Dagobert Soergel. The art and architecture thesaurus (AAT): A critical appraisal. *Visual Resources*, 10:369–400, 1995.

[Sow84] John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley, Reading, MA, 1984.

[Sow87] John F. Sowa. Semantic networks. In Shapiro and Eckroth [SE87].

[SR92] Stuart C. Shapiro and William J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5):243–275, January–March 1992. Also in [Leh92].

[SSGC89] Philip J. Smith, Steven J. Shute, Deb Galdes, and Mark H. Chignell. Knowledge-based search tactics for an intelligent intermediary system. *ACM Transactions on Information Systems*, 7(3):246–270, 1989.

[Sto84] Michael Stonebraker. Adding semantic knowledge to a relational database system. In Brodie et al. [BMS84].

[Sto88] Michael Stonebraker, editor. *Readings in Database Systems*. Morgan Kaufmann, San Mateo, CA., 1988.

[Sve89] Elaine Svenonius. Design of controlled vocabularies. In *Encyclopedia of Library and Information Science*. Marcel Dekker, 1989.

[Tar92] Zahir Tari. Designing object-oriented databases with a semantic data model and a rule model. In *Database and Expert Systems Applications, Proceedings of the International Conference in Valencia, Spain*, pages 108–114. Springer-Verlag, 1992.

[ULA94] *Union List of Artist Names — User's Guide to the Authority Reference Tool, Version 1.0*, Getty Art History Information Program, G.K. Hall, New York, 1994.

[WD91] Edgar B. Wendlandt and James R. Driscoll. Incorporating a semantic analysis into a document retrieval strategy. In *SIGIR Proceedings of the 14th Conference an Research and Development in Information Retrieval, Chicago*. ACM, 1991.

[Woo75] William A. Woods. What's in a link: Foundations for semantic networks. In D.G. Bobrow and A.M. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–82. Academic Press, 1975. Also in [BL85].

[ZM89] Stanley B. Zdonik and David Maier, editors. *Readings in Object Oriented Database Systems*. Morgan Kaufmann, San Mateo, CA, 1989.