Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

**Research Report**
RR-91-17

# The Use of Abstraction Concepts for Representing and Structuring Documents

**Andreas Dengel & Nelson M. Mattos**

**July 1991**

**Deutsches Forschungszentrum für Künstliche Intelligenz GmbH**

Postfach 20 80
D-6750 Kaiserslautern
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

# Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Nixdorf, Philips and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

# The Use of Abstraction Concepts for Representing and Structuring Documents

**Andreas Dengel & Nelson M. Mattos**

DFKI-RR-91-17

This report is an extented version of a paper [MD91] that has been published in the proceedings of the Fourth International Symposium on Artificial Intelligence, Cancún, Mexico, November 1991.

# The Use of Abstraction Concepts for Representing and Structuring Documents

ANDREAS DENGEL & NELSON M. MATTOS[1]

**Authors´ abstract**

Due to the amount of documents available in modern offices, it is necessary to provide a multitude of methods for the structuring of knowledge, i.e., abstraction concepts. In order to achieve their uniform representation, such concepts should be considered in an integrated fashion to allow concise descriptions free of redundancy. In this paper, we present our approach towards an integration of methods of knowledge structuring. For this purpose, our view of abstraction concepts is briefly introduced using examples of the document world and compared with some existing systems. The main focus of this paper is to show the applicability of an integration of these abstraction concepts as well as their built-in reasoning facilities in supporting document processing and management.

**CONTENTS**:



[1] University of Kaiserslautern, CS Department,
Erwin-Schrödinger-Straße, D-6750 Kaiserslautern, Fed. Rep. of Germany

## 1 INTRODUCTION

All activities in an organization require or produce information. The future in the office domain will be characterized by new fundamental tendencies that lie not only in a more comprehensive possibility for information representation and interchange, but also in an altered processing. As the main information carrier, documents are the central aid for the integration of office functions [Do85]. In offices, documents arrive in varieties of forms: paper, electronic signals, sounds, or pictures. A person creates a document, and later on, another person attempts to interpret the acquired information in order to extract the intended meaning. Due to the amount of information to be processed and the multitude of different kinds of documents (printed purchase orders or articles, E-mail, or News), there is a pressing need for intelligent document analysis systems that should lead to a full understanding of the captured information. For this purpose, it is necessary to develop mechanisms for an automatic document analysis as well as formalisms allowing the description of the knowledge resulting from an analysis and the interrelations between portions of it.

The task of automatic document structure extraction will not be considered in this paper but have been presented in earlier publications [DB88, DB89]. For document representation, several formalisms provide distinct concepts with respect to the different views of a document (layout view, logical view, contents). They range from word-based representation [Fa85] and descrip-tors, [SM83] to techniques for structural representation [EK89, DM90].

When considering structural representation of documents, constructs for their overall organization in the office environment play a key role [DM90]. Not surprisingly, most of these constructs have their roots in epistemological methods for structuring knowledge, i.e., in trying to keep track of the amount of information existing in the office domain, people usually involuntarily apply some *abstractions* in order to structure documents in some desired form [Ma88a]. Abstraction concepts (i.e., classification, generalization, association, and aggregation) and their underlying relationships [BMW84, Br81, Ma88a, SS77] provide natural ways to structure or organize knowledge portions in the form of objects. They enable specific details of particular objects to be suppressed and those pertinent to the view of information at hand to be emphasized. Moreover, they provide reasoning facilities [Ma88a, RHMD87] that can be exploited by a knowledge model to make deductions about objects as well as to guarantee the structural and semantic integrity of their representation. Therefore, they are a fundamental construct for organizing knowledge in application areas requiring a high degree of information structuring, like in the office domain.

In the last few years, many models [BS85, HM78, PM88, HK87] have been developed following the semantics of one or another abstraction concept. Nevertheless, approaches towards an integration of all of them are rarely found. Most existing models concentrate on the support of one abstraction concept, not taking the existence on the others into account. This weakens the expressiveness and the semantic power of such models and forces the application to perform a cumbersome modeling in order to be able to represent the concepts not supported by a system. For this reason, we argue that object descriptions should embody all abstraction concepts so that objects of the model, like the real world entities, can play different roles at the same time (i.e., class, aggregate, instance, etc.), depending only on the relationships they have to other objects. Note, for example, that an object such as a concrete letter is at the same time an instance of the class letters, an aggregate composed of the several layout objects such as text-blocks, lines, words, and characters, and an element of top-secret documents. This integrated view of abstraction concepts, which has been presented in [Ma88a], was incorporated into the knowledge model of a prototypical Knowledge Base Management System (KBMS), called KRISYS [KR89,Ma88b]. The knowledge model of KRISYS supports an object-centered representation of a domain, putting special emphasis on a complete support of abstractions concepts in order to use their semantics as the basis for drawing conclusions about objects and for maintaining the integrity of the knowledge base. This provides an accurate and reliable representation of all information structures encountered in document processing. Such an integrated view of abstraction concepts is one of the aspects that differentiate KRISYS from existing knowledge engineering tools like ART [C185], KEE [FK85], KNOWLEDGE CRAFT [FWA85], LOOPS [BS83, SB86], etc. as well as from existing database systems [Da87].

In this paper, we discuss our approach towards an integration of these abstraction concepts. We briefly introduce our view of abstraction concepts, illustrating their meaning by examples of the document world. Then, we demonstrate their applicability by examining their built-in reasoning facilities and show how an integration of abstraction concepts can be exploited to keep track of the pressing need to support human clerks in information processing and management. Finally, we compare our approach with some existing systems and provide an outlook to future and ongoing research activities.

## 2 MODELLING DOCUMENTS WITH ABSTRACTION CONCEPTS

In general, an application world can be seen in terms of entities (i.e., *objects)* having descriptions (i.e., *properties* and *relationships* to other objects) and *constraints* to distinguish ´reality´ from other possible worlds* [Bo86]. For example, in our domain, different kinds of documents exist, and each of them has properties such as generation-date, purpose, etc. Domain dependent relationships can also be specified to allow objects to carry some further information. For example, a document is related to some persons who have generated it. However, there are important domain independent relationships that occur in nearly every domain and as such have well defined semantics. They are the relationships underlying the several abstraction concepts [BMW84,Br81,Ma88a,SS77].

In KRISYS, every entity existing in the application domain is expressed as an object of its model, the so-called *schema* [Kr89,Ma88b]. A schema is the symbolic representation of a real world entity, roughly analogous to frame or unit in other representation systems. It is always identifiable by a unique schema name and is composed of a set of attributes. The attributes may again be further described by aspects expressing integrity constraints (e.g., possible-values, cardinality restrictions) or other specifications that characterize an attribute in more detail (e.g., default-values, comment). Attributes represent either properties or relationships of real world entities. They may be of different types depending on their underlying semantics, i.e., whether they express domain specific properties and relationships or abstraction relationships.

---

*The term object is used here to refer to a data structure in the knowledge base that is intended to denote an entity in the world. For example, a knowledge base might contain an object to represent the letter of Mr. White to Mrs. Brown.

## 2.1 AGGREGATION

The abstraction concept of aggregation relates objects in order to build more complex ones. That is, it treats objects not as atomic entities, but as a composition of other objects. In the office environment, documents are characterized by their layout structure and logical structure [Ho85]. Both represent hierarchical structures, but provide a different view of the same contents. The layout structure divides the contents of a document into hierarchically nested layout objects. It is, therefore, an aggregation hierarchy reflecting nested physical parts of information, the so-called layout objects. For example, a document is composed of pages containing several graphic- and text-blocks. These are, in turn, respectively composed of several graphic primitives and text-lines. The latter ones are built of words that are composed of characters. The logical structure divides the contents of a document into a hierarchy of logical objects. Thus, it is also an aggregation of conceptual parts of information, called logical objects. For example, the letter presented in Figure 1 has logical objects like logo, sender, subscript, footnote, etc. Subscript is, in turn, divided into subscriber, signature, and regards.

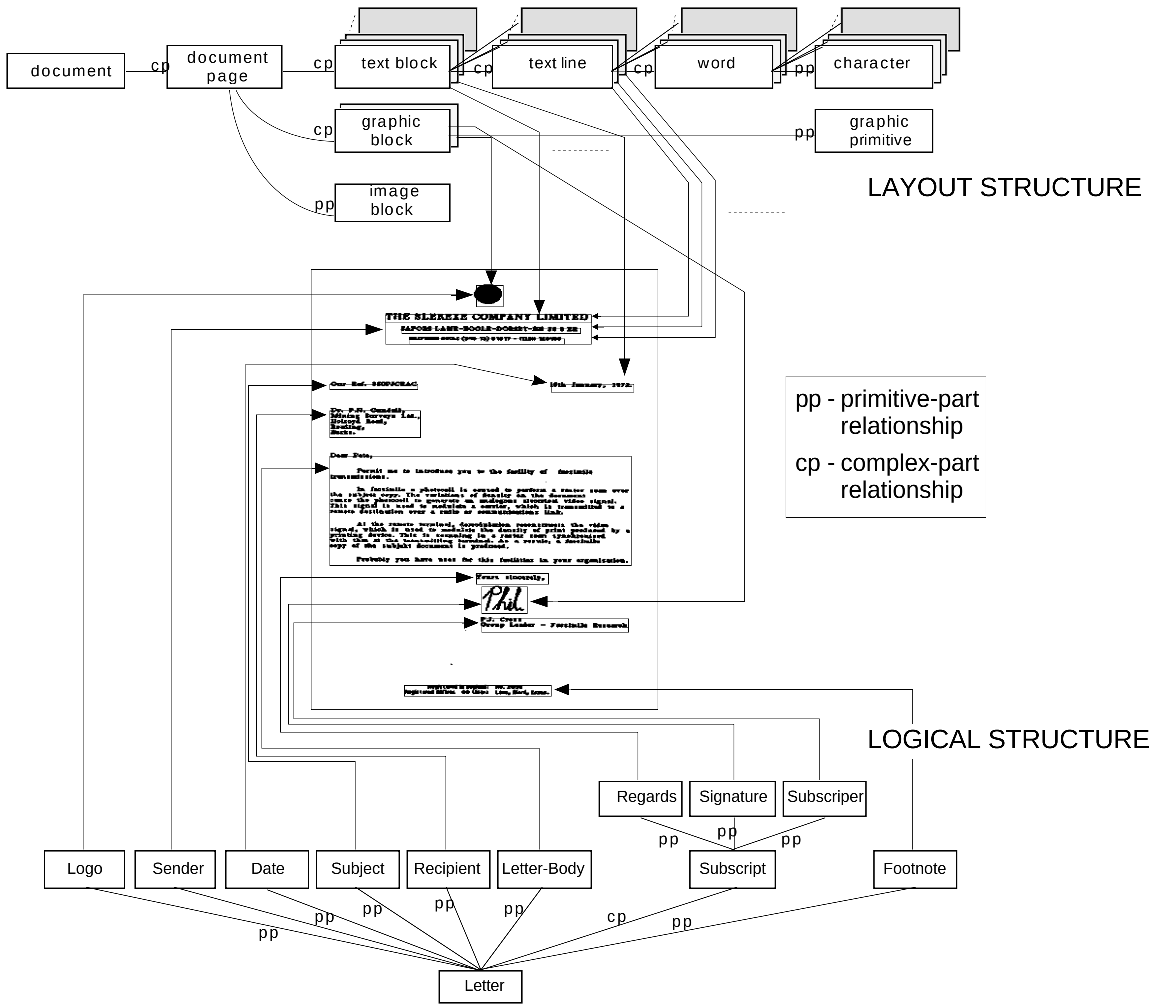


**Figure 1**: Layout Structure and Logical Structure of a Business Letter up to the Block Level.

The international standard ODA [IS085] (office document architecture) provides collections of rules for defining layout and logical structures, but with very restricted aggregation semantics. This is similar to knowledge representation systems enabling a specification of references between objects. In these systems, such references do not embody any special semantics, serving only as flexible means for the representation of any kind of relationships between objects. Aggregation relationships (or part-of relations), on the other hand, have a very specific meaning. They represent the idea that an object 'consist-of' other objects, thereby expressing that an object cannot 'consistently' exist without its parts [Ma88a]. Additionally, it is important to observe that aggregation itself differentiate between two kinds of objects and consequently of relationships. Primitive-parts express atomic objects (e.g., characters and graphic primitives), i.e., those parts which cannot be further decomposed. Together, they constitute the lowest-level complex-parts or simplest aggregates. These can, in turn, be used to build more complex higher-level parts so that an aggregation or part-of hierarchy is established. Therefore, it is necessary to observe the importance of modeling aggregates by means of special-purpose relationships, differentiating between complex- and primitive-parts.

For these reasons, the aggregation concept is expressed in KRISYS by special-purpose attributes specified by the user which can be of two kinds: 'primitive' and 'complex'. Attributes of kind 'primitive' represent relationships between a complex and an 'atomic' primitive-part, whereas 'complex' attributes relate two complex-parts (see also Figure 1). In order to guarantee a symmetric modeling, each aggregation relationship will correspond to a pair of symmetric KRISYS attributes expressed by the aspect diametric-reference, i.e., to an attribute of kind complex-up there is another of type complex-down and to a primitive-up attribute there is another of kind primitivedown. Figure 2 shows an example of a layout aggregation including the representation of the symmetric attributes.

document page

**in-document** complex-up, ...
possible-values (instance-of (document))
cardinality [ 1 ∞ ]
diametric-reference (has-doc-page)

**has-text-block** complex-down, ...
possible-values (instance-of (text block))
cardinality [ 1 ∞ ]
diametric-reference (in-doc-page)

**has-graphic-block** complex-down, ...
possible-values (instance-of (graphic block))
cardinality [ 0 4 ]
diametric-reference (in-doc-page)

graphic-block

**in-doc-page** complex-up, ...
possible-values (instance-of (document page))
cardinality [ 1 1 ]
diametric-reference (has-graphic-block)

**has-graphic-primitive** primitive-down, ...
possible-values (instance-of (graphic-primitive))
cardinality [ 1 ∞ ]
diametric-reference (in-graphic-block)

**Figure 2**: Examples for the Aspect-Diametric References in KRISYS.

Moreover, since the aggregation concept is represented in KRISYS by means of user-defined attributes (of the kinds mentioned above), it allows the specification of several aggregation relationships between objects, each of which with very fine semantics (observe, for example, the distinct integrity constraints associated to the attributes 'has-text-block' and 'has-graphic-block', which cannot be expressed by systems supporting aggregation by means of one single part-of relationship).

Another aspect of KRISYS is the differentiation between dependent and independent parts in the semantics of its operations [KBG89]. In the case of a dependent relation, for example, the deletion of an aggregate will recursively remove all its parts, freeing the application from the search and subsequent deletion of all parts of the removed object. Thus, the existence of dependent parts directly depends on the existence of its aggregates.

However, dependent relations do not allow the reusing of objects in another aggregate in a later time, which would be certainly desirable in some cases. Assume, for example, that facsimiles within documents are treated as being independent. In such a case, the deletion of the corresponding document in which the facsimile appears does not force a deletion of the facsimile itself so that it can be kept separately within a special image base. Therefore, the aggregation concept in KRISYS provides means for the definition of high-structured objects, allowing a user at a particular moment to view such structured object as a single unit (i.e., together with its parts) and at a subsequent time to view only parts of it as independent or dependent objects. So, objects can be treated as objects in their own right or as components of other objects. When observing such components, one can further differentiate between *exclusiue* or *shared components* [KBG89]. This can be, for example, applied for having different access authorizations to portions of information. In other words, some documents or document parts are public (shared), while others are primary personal (exclusive). Actually, the kind of components supported by the model determines the form of allowed aggregation hierarchies. Tree-like hierarchies provide only for exclusive components, whereas network-like ones also for aggregates with common/shared parts.

## 2.2 CLASSIFICATION AND GENERALIZATION

Found in almost every existing data or knowledge representation model, classification is the best understood form of abstraction. It provides an important means for organizing the application world by allowing the modeler to refer to a class as a representative or prototype of its instances, into which a specification of both properties and constraints applicable to all instances is presented. Generalization complements the classification concept by allowing the definition of superclasses.

In KRISYS (and in many other knowledge representation systems), standard attributes occurring in each schema are used to model the abstraction concepts of classification and generalization: has-instances and instance-of for classification and has-subclasses and subclass-of for generalization (Figure 3).

However, since we treat any abstraction relationship as an object attribute (as illustrated for aggregation in Figure 2), KRISYS allows the definition of constraints for such relationships, e.g., every instance of letter cannot be an instance either of report or of message.

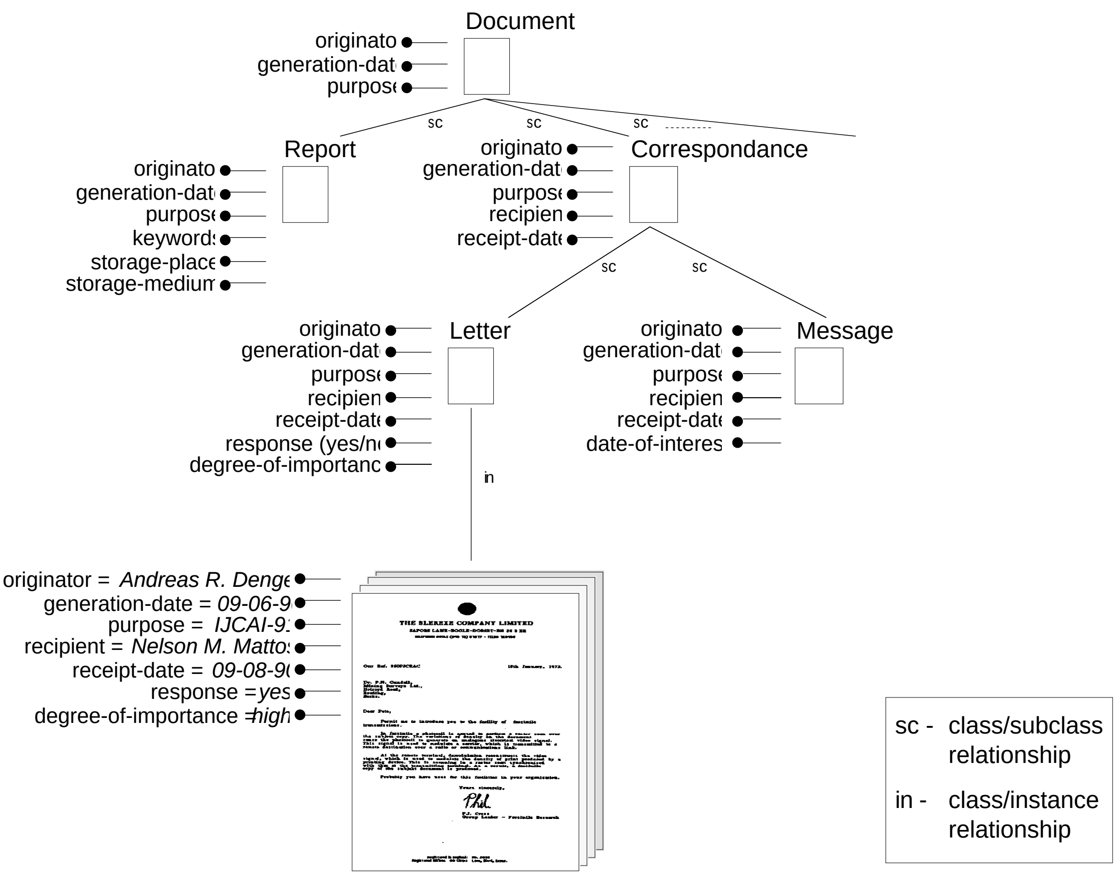


**Figure 3**: Generalization Hierarchy of Document Types.

## 2.3 ASSOCIATION

In the abstraction concept of association, objects, so-called elements, are considered as a higher level object, called set, e.g., heterogeneous documents like a request, an offer, an order, and an invoice may belong to the same procedure. The details of the element objects are suppressed and the properties of the group are emphasized in the set object, e.g., whether the procedure is in-time or not. Like the other abstraction concepts, association also relates objects by means of subset-of relationships building an association hierarchy.

Clearly, one does not group any objects together. In general, only objects fulfilling common conditions are grouped into sets by association. For this reason, in addition to set properties (i.e., the characteristics of a set as a whole), the description of set objects in KRISYS contains the so-called membership stipulations, expressing the necessary conditions that have to be satisfied by objects in order to become elements of the set [Ma88a]. This refined semantics of sets can be flexibly employed to classify documents according to different criteria. Figure 4 illustrates the example of submitting a paper to IJCAI-91. Here, different documents, like a 'call-for-paper', the correspondence between the authors of a joint paper as well as the entire paper are associated to one and the same procedure 'IJCA1-91', because they fulfill the membership stipulation of this procedure. At the same time, this procedure has set properties expressing its actual state (e.g., paper written or in preparation), whether it is in-time or not, etc.

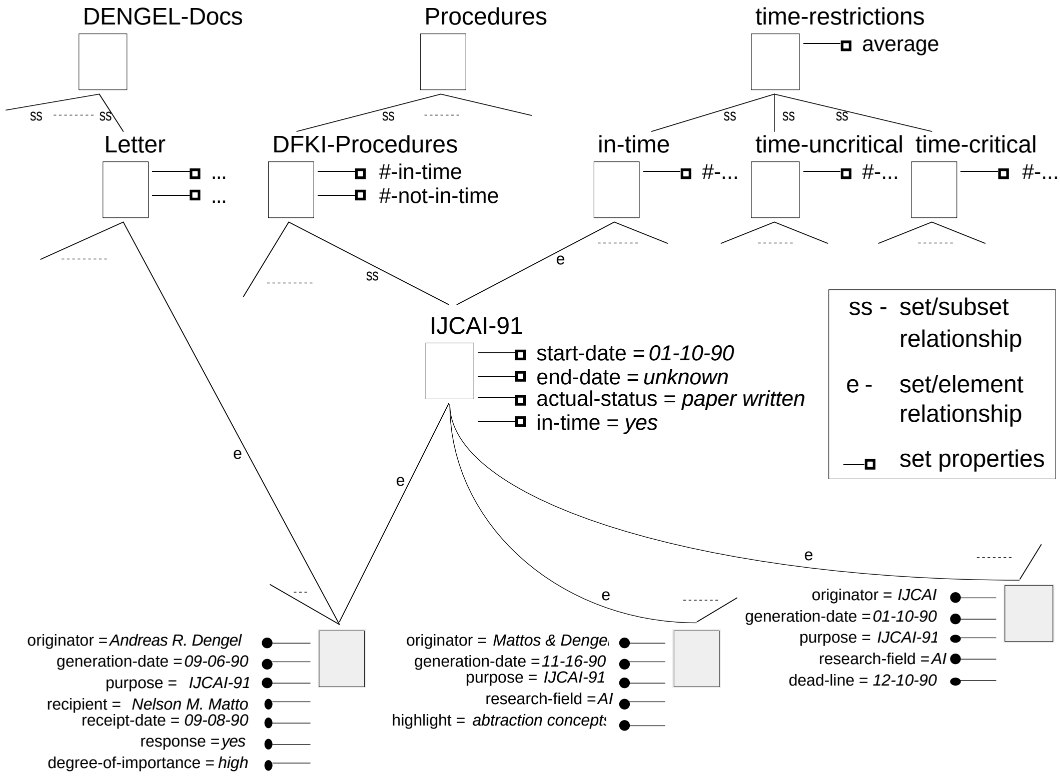


**Figure 4**: .Example for Associating Heterogeneous Documents.

Here, the difference between generalization/classification and association should be clear. Classes define the structures (i.e., attributes and related integrity constraints) of the instances, while sets do not. Indeed, sets may group heterogeneous objects together since the specification of the membership stipulation does not necessarily depend on the structure of the elements. Note that in the example, all objects which are related to IJCAI and generated in 1990 or 1991 are grouped in the set procedure 'IJCA1-91' (independent of their structure). Thus, in the association concept there is no inheritance!

## 3 BUILT-IN REASONING FACILITIES

Once, the semantics of the abstraction concepts are fully understood by a system, it is then able to undertake some tasks automatically, e.g., to draw particular conclusions about objects and to keep the knowledge base in a consistent state, by exploiting the so-called built-in reasoning facilities. For example, based on the notion of transitivity inherent in the abstraction concepts (e.g., parts of a component are also parts of the component's supercomponents), the introduction of cycles or other ambiguities into a hierarchy clearly contradicts the meaning of the abstraction concepts, and are therefore prevented by KRISYS.

Similarly, based on the representation of concepts by means of a pair of symmetric attributes, on modifications of abstraction hierarchies, the referential integrity of such relationships is automatically maintained by the system. This can cause, for example, the corresponding inverse attributes to be immediately updated, but permits also the support of a refined delete semantics, considering, for example, the kind of components of an aggregate (i.e., dependent/independent). It additionally enables a precise control of allowed connect and disconnect operations based of the definition of shared/exclusive objects.

### 3.1 AGGREGATION SPECIFIC REASONING

When analyzing an aggregation hierarchy, one may notice that there are properties of objects which can be characterized as monotonic [Ma88a]. For example, by expressing that the value of a particular property is upward implied (e.g., extensions of a layout object), the system may control whether it increases going upward over the aggregation hierarchy, thereby preventing the violation of such monotony or triggering further changes in the knowledge base to keep the truth of the predicate (whenever possible). Such characteristics can be described in KRISYS by means of so-called implied-predicates.

### 3.2 CLASSIFICATION/GENERALIZATION SPECIFIC REASONING

The reasoning mechanism provided by the generalization/classification hierarchy is inheritance. Since classes define the structure (i.e., attributes and constraints) of their instances and subclasses, by inserting objects in the knowledge base and expressing the belief of the modeler that they are instances or subclasses of some classes, KRISYS can reason that they have the properties defined by these classes.

However, objects may exist in the knowledge base *without the specification of a class*. Frequently, the user knows about the existence of an object but not of its class so that it has to be first introduced in the knowledge base without any instance-of specification. After this, the user might reflect upon its class or he might want to see how the object would look if it were an instance of a particular class (e.g., he 'believes' at first that a document is a private message) and defines an instance-of relationship. (Note that in this case the user is using the system as a tool to determine the structure of the objects, by dynamically defining instance-of relationships. This corresponds to the idea of 'discovering' and not 'inventing' abstractions as presented by [JF88].) KRISYS will then deduce the object structure, generating a more detailed description of this object. Based on this new description, the user might now realize that it does not correspond to the real world entity, starting the determination of the class once again (i.e., he will realize that the document is not a private message but a business letter). Therefore, instance-of relationships in KRISYS correspond, in truth, to current beliefs of the user, which may change at any time without affecting the existence of other objects or further descriptions (i.e., properties, constraints, etc.) that an object may possess. (Obviously, the same holds for objects connected by subclass-of relationships).

### 3.3 ASSOCIATION SPECIFIC REASONING

Reasoning on association hierarchies is at first provided by means of the membership stipulations. Since KRISYS guarantees that every element of a set always fulfils the corresponding membership stipulation, it is possible to deduce, for example, that all elements of the procedure 'IJCA1-91' satisfiy the condition of being related to the IJCAI conference and generated in either 1990 or 1991. KRISYS may also reject operations that connect an element to a particular set whenever such objects do not fulfil the membership stipulations of the set. Furthermore, KRISYS can determine whether a change in an element should cause the dissolution of an existing association relationship or the creation of a new one because of the satisfaction of the membership stipulation of another set. As a consequence, in the example of Figure 4, when an author sends a letter to another author on September 6th, 1990 with the purpose of defining the contents of a joint paper for an IJCAI conference, KRISYS can automatically associate this document with the set representing the procedure 'IJCA1-91'. Similar conclusions can be drawn when modifying membership stipulations. In this case, not only may elements be disconnected from a particular set, but the set itself might be dissolved from its relationships to supersets since an object may be a subset of another only if it possesses more restricted membership stipulations*.

Further reasoning capabilities are provided by means of set properties. Since such set properties are in general based on characteristics of each individual element, conclusions about the values of set properties can be drawn from the elements' attributes. So, upon changes to elements (e.g., because of a modify operation) or on the relationships between elements and sets (e.g., because of the introduction or deletion of an element), the recalculation of the values of set properties is also performed by KRISYS automatically. As such, when KRISYS associates the paper to the set 'IJCA1-91', it can automatically modify the actual state of this set to 'paper written' and determine that the procedure 'IJCA1-91' is now in-time. This will, in turn, cause the dissolution of the relationship between the procedure 'IJCA1-91' and time-critical procedures and its inclusion in the set of in-time procedures, thereby provoking changes on the numbers of time-critical and in-time procedures as well as a recalculation of the average of the time-critical procedures (see Figure 4).

# 4 INTEGRATED VIEW OF ABSTRACTION CONCEPTS

## 4.1 INTEGRATING DIFFERENT ABSTRACTION ROLES OF A SAME CONCEPT

As already mentioned in the introduction, an object in KRISYS does not play an abstraction role (i.e., class, aggregate, part-element, set, etc.) by itself, but only in the context of being related to other objects. Consequently, the semantic meaning of objects can be different from context to context in which they are found. In the case of aggregation, this means that there can be objects expressing an aggregate (i.e., something that can be decomposed) in one context and a primitive part (i.e., an atomic part) in another. Thus, the semantic meaning of an object (i.e., its role) can only be determined by considering a particular context. Think about the automatic recognition of the form of a document as a whole and its proper identification as a memorandum, personal letter, or medical record. This gross level of document recognition requires only the abiltity to define types of documents by their geometric block format [DB88]. From here, it is a short step to link, for example, portions of text that are contained in layout objects to specific logical objects, identifying blocks in which the agency would be named, and blocks in which the applicant would be named, etc. At this level, it is enough to consider logical objects, like the sender or the subscript of a letter as atomic parts. For a further analysis of the document, it seems to be appropriate to consider the textual information within several logical objects, for example, to verify the originator (sender of the letter) as the subscriber. In such a case, the sender object itself may represent in another context an aggregation consisting of name and address that are further refined. This allows, for example, a system like KRISYS to provide a view of information respecting the analysis depth being performed.

---

* The current version of KRISYS does not provide an automatic definition and dissolution of subset-of relationships, but only of element-of relationships. This is subject of our ongoing research activities.

The integration of several classification/generalization roles into one object permits the representation of different meta-levels corresponding to distinct views of such an object in the application domain. In this case, KRISYS guarantees that the different semantics associated with each of these meta-levels are separately maintained from one another, so that attributes of one level are not passed on further to the other level.

Also in the association, an object can play different roles at the same time. For example, the object 'IJCAI-91' procedure in Figure 4 is a set of heterogeneous documents and an element of time-critical procedures. Note that such an integration is necessary in this context because the procedure I'JCAI-91' itself and not its documents is time-critical.

### 4.2 COMPLETE INTEGRATION

Up till now, we have isolated the discussion about each abstraction concept from the others. We have seen, for example, that association builds a hierarchy of objects which, however, up to this point, was constructed independently of the generalization or aggregation hierarchy. Naturally, in the real world, this independence does not exist. As illustrated in Figure 5, aggregation as well as generalization/classification concepts occur in an integrated fashion. Letters may be specialized according to the kinds of layout and logical objects they possess. Moreover, objects are obviously not only instances of classes and parts of aggregates, but also elements of sets. For example, the particular letter from Mr. Dengel to Mr. Mattos on September 6th, 1990 is at the same time an instance of business letters, an aggregate composed of a sender, date, letter-body, subscript, etc., and an element of the set of documents of the IJCA1'91 procedure. This integrated view is shown in Figure 6.

Hence, objects in KRISYS can play up to six roles at the same time, depending only on the types of relationships to other objects (instance, subclass, element, subset, primitive-part, complex-part). Obviously, when an object represents both a set and a class, the set properties should not affect the instances, and the instance properties should not be used for describing the set. In an analogous way, when combining aggregation with the other concepts, instances and set properties should not be confused with those used to express the aggregation (i.e., the attributes of type 'complex' and 'primitive'). The latter ones are different from the other properties of the objects (i.e., those concerned with classification/generalization and association) since their values are to be interpreted as other objects of the knowledge base. In other words, while aggregation properties represent parts of the objects, set and instance properties express characteristics of them. KRISYS guarantees a clear differentiation of these kinds of properties by means of a refined specification of attribute types. Such a mechanism has been described in [MM89].

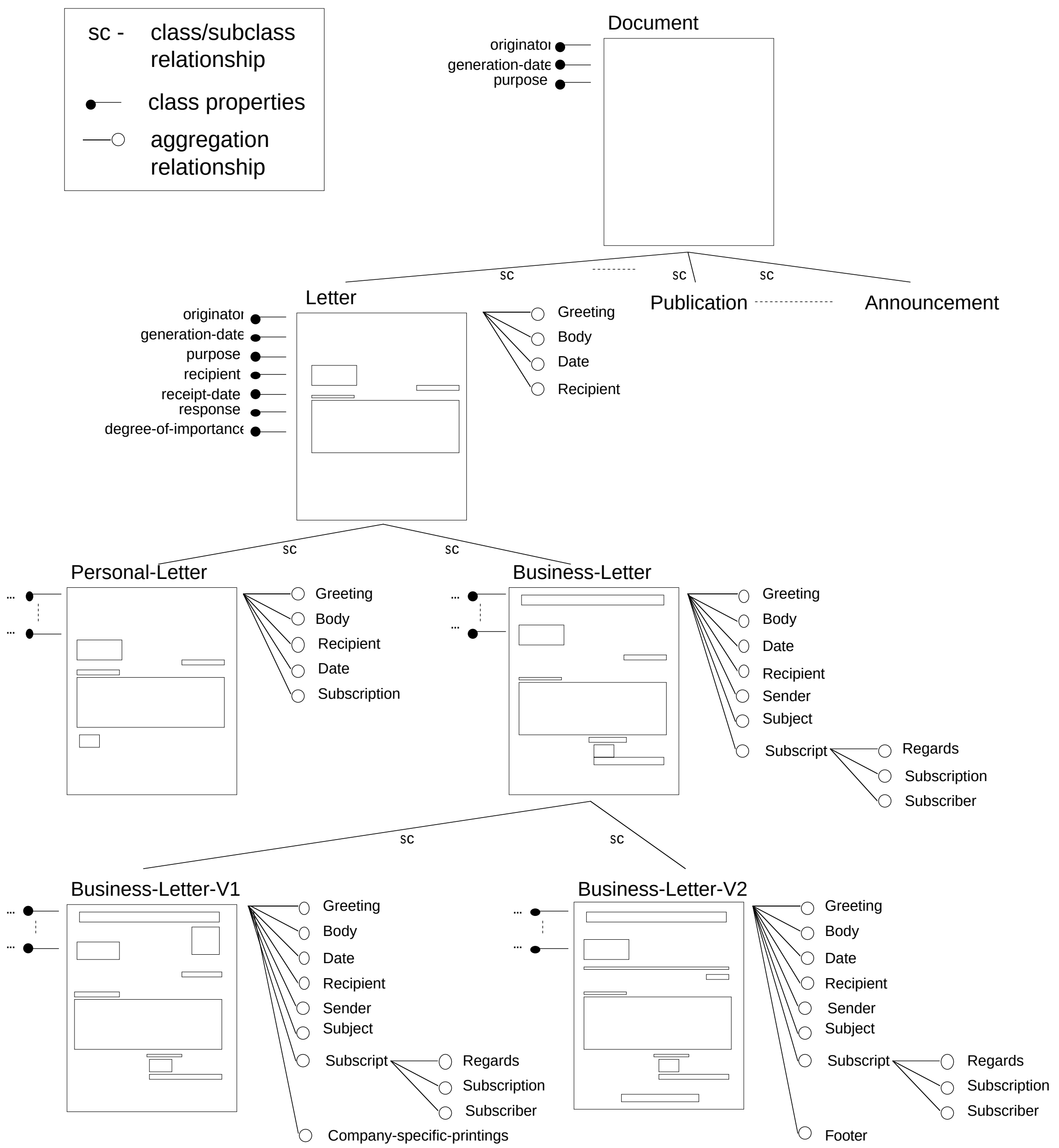


**Figure 5**: Integration of Aggregation and Generalization/Classification Concepts for Document Representation.

To emphasize our ideas, we would like to briefly illustrate how the integrated view of abstraction concepts can undertake some of the tasks in document processing. Considering the office domain, the task of document processing and representation amounts to a structural as well as a conceptual analysis to obtain a multitude of information possibilities. In a first analysis step, the layout structure of a document at hand has to be established [De90]. As already mentioned, this is a treelike aggregation that offers easy modification and efficient access down to atomic part-elements (e.g., characters) following the part-relationships within the hierarchy. In the next step, a logical labeling has to be applied. Thereby, layout objects, or groups of them, are assigned a label that designates a corresponding logical object (e.g., the date or the recipient of a letter). For this purpose, we use a special generalization/classification hierarchy [DB88]. It defines document classes having several arrangements of logical objects on different abstraction levels (see Figure 5). Note again that we do not want to describe the entire document analysis procedure, but rather concentrate on aspects of internal representation. While logical labeling is applied, the information arrangement of a document at hand is matched against the different document classes of the hierarchy, thereby passing through it in a top-down manner. As a result, a logical structure is established which is expressed by an aggregation hierarchy. In Figure 5, this is illustrated by the aggregation relationships assigned to document classes (see also Figure 1).

In each step through the tree, an instance-of relationship is generated to the class, the document at hand is actually matched against. When defining an instance-of relationship, KRISYS will deduce the corresponding object structure, generating a more detailed description of this object. Thus, each step forces the creation of an instance, while another one has to be deleted (i.e., the one of its superclass). This is supported by dynamical definition of instance-of relation-ships provided by KRISYS (see Chapter 3.2).

As illustrated in Figure 6, corresponding instance properties have to be filled after every instance-of relationship definition. For this reason, (Optical Charcater Recognition in the case of paper documents [HBD90] and) keyword analysis is initiated. Every logical labeling of layout objects activates a corresponding demon (i.e., an attached procedure) which is automatically triggered if a value (i.e., a corresponding text block) is assigned to it. The purpose of such a demon is to activate a special package of rules which then will interpret the contents of the text blocks extracting its keywords. These keywords are the basis for a dynamic definition of element-of relationships. That is, they are used, for example, to deduce that the letter from Mr. Dengel to Mr. Mattos sattisfies the membership stipulation of the IJCAI-91 procedure, and therefore belongs to it. Thus, we are able to provide a representation of documents that is based on an integration of aggregation and generalization/classification as well as association, like shown in Figure 6.

In summary, when modeling an application world, it is important to consider all abstraction concepts together so that all information about an entity is concentrated into one object of the model. This has an additional advantage. By considering all abstraction concepts together, descriptions of one concept can use the descriptions of the others, thereby allowing much richer and more precise descriptions of the world entities to be defined. A final advantage is that objects can be concisely described without losing the necessary flexibility, and some classical errors provoked by the introduction of redundancy can be avoided by reducing the amount of repetition in the descriptions. Note that there is no redundant storage of document objects. For a further analysis, only the definition of additional relationships is necessary.

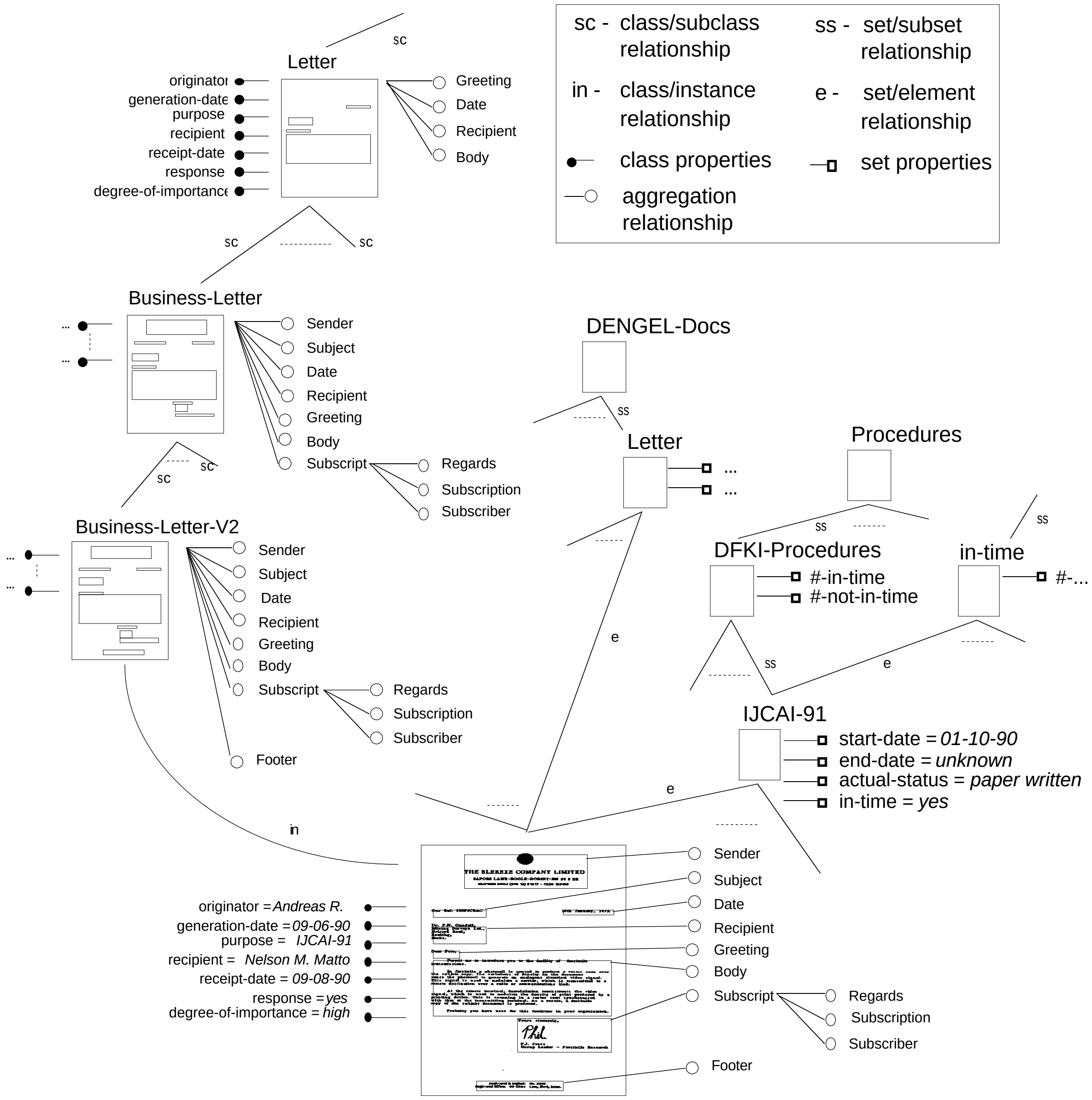


**Figure 6**: Integration of Aggregation and Generalization/Classification as well as Association Concepts for Document Representation.

## 5 COMPARISON TO OTHER SYSTEMS

The support of all abstraction concepts in an integrated fashion is one of the most important aspects that differentiate KRISYS from systems like ART [C185], KEE [FK85], KNOWLEDGE CRAFT [FWA85], LOOPS [BS83,SB86], etc. as well as from database systems. These systems neglect the existence of some of these concepts (generalization by database systems, association by database systems, KEE, and LOOPS, and aggregation by all of them), thereby severely weakening the expressiveness and the semantic power of their knowledge or data model. Furthermore, since KRISYS incorporates all roles of a real world entity in one single object, there is no need to introduce two distinct representations to support both association and generalization/classification (as in the case of ART and KNOWLEDGE CRAFT). It is also not necessary to make a kind of 'hodgepodge' with the semantics of the generalization/classification in order to be able to support the representation of set properties (as done by KEE). Following the lines of its integrated view of knowledge base objects, KRISYS does not forces a substantial amount of real world semantics to be maintained in the application programs, nor the introduction of redundancy into the knowledge base. Even proposed extensions of database technology (semantic data models [HK87,PM88], object-oriented data models [Di88], etc.) focus only on some of these concepts (mostly aggregation and/or generalization/classification) and neglect most of the underlying reasoning facilities.

Finally, KRISYS treats the abstraction concepts as dynamic relations, which may be changed by the user or the application at any time. The corresponding built-in reasoning is, in such cases, automatically applied, keeping the knowledge base in a semantically consistent state. Inheritance, for example, is in some systems (ART and LOOPS) not more than a means to save some typing during the modeling process. Changes in the structure of a class (deletion, creation of attributes, etc.) are either not allowed or not reflected in the structure of the existing instances until 'compilation', leading to severe inconsistencies. In KRISYS, every time that relevant information is changed, inheritance as well as the other built-in reasoning facilities are evaluated again so that the system guarantees the structural and semantic integrity of the knowledge base at any time.

## 6 SUMMARY AND OUTLOOK

In this paper, we have discussed abstraction concepts and their built-in reasoning facilities in representing and structuring documents. The focus of the paper has primarily been on defining the semantics of the several concepts and their integration to show their applicabiliy for the office domain. Having a document analysis system, as proposed in [DB89, HBD90], that is capable of revealing the structure as well as keywords of the contents of a document at hand, the ideas described in this paper represent an excellent basis to support a human clerk, making his tasks in information processing easier.

We have argued that object descriptions must embody all abstraction concepts, so that objects can play different roles at the same time depending on the context. Based on the relationships that express a context, deductions by means of their built-in reasoning facilities, the most important advantage of abstraction, can be performed. However, further work is necessary. For example, the representation of time semantics for a version management, or for deducing the order in a sequence of documents that describe a procedure. Other future considerations are a study of space or null values (e.g., not existing, unknown) in the context of abstraction concepts.

## REFERENCES


[BMW84] Borgida, A., Mylopoulos & J., Wong, H.K.T.: *Generalization/Specialization as a Basis for Software Specification*, in: Brodie, M.L., Mylopoulos, J., Schmidt, J.W. (eds.): On Conceptual Modelling, Springer-Verlag, New York (1984), pp. 87-114.

[Bo86] Borgida, A.: *Survey of Conceptual Modeling of Information Systems*, in: Brodie, M.L., Mylopoulos, J. (eds.): On Knowledge Base Management Systems, Springer-Verlag, New York (1986), pp. 461-470.

[Br81] Brodie, M.L.: *Association: A Database Abstraction for Semantic Modelling*, in: Proc. 2nd Int. Entity-Relationship Conference, Washington, D.C. (1981).

[BS83] Bobrow, D.G. & Stefik, M.: *The LOOPS Manual*, Xerox PARC, Palo Alto, CA (1983).

[BS85] Brachman, R.J. & Schmolze, J.G.: *An Overview of the KL-ONE Knowledge Representation System*, in: Cognitive Science, Vol. 9, No. 2,1985, pp. 171-216.

[Cl85] Clayton, B.D.: *Inference ART Programming Tutorial,* Volume 1 Elementary ART Programming, Volume 2 A First Look At Viewpoints, Volume 3 Advanced Topics in ART, Los Angeles, CA (1985).

[Da87] Date, C.J.: *An Introduction to Database Systems,* Vol. 1 and Vol. 2, 4th edition, Addison-Wesley Publishing Company, Reading, Mass. (1987).

[DB88] Dengel, A. & Barth, G.: *High Level Document Analysis Guided by Geometric Aspects*, Internat. Journal on Pattern Recognition and AI, Vol. 2, No. 4 (1988), pp. 641-656.

[DB89] Dengel , A. & Barth, G.: *ANASTASIL: A Hybrid Knowledge-based System for Document Layout Analysis*, Proceedings of the 11th IJCAI, Detroit, MI (1989), pp 1249-1254

[De90] Dengel, A.: D*ocument Image Analysis - Expectation-Driven Textual Recognition*, Proceedings of the SSPR90 (Interntl. Association for Pattern Recognition Workshop on Syntactic & Structural Pattern Recognition, Murray Hill, NJ (1990), pp. 78-87

[Di88] Dittrich, K.R.: *Advances in Object-Oriented Database Systems* (Proc. of the 2nd Int. Workshop on ObjectOriented Database Systems), Lecture Notes in Computer Science 334, Springer-Verlag, Berlin (1988).

[DM90] Dengel, A. & Mattos, N.M.: *Integration of Document Representation, Processing and Management*, DFKI Research Report RR-90-03, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern Site (1990), 18 pages, also in Proceedings COIS-90, Boston, MA, April 1990, pp. 118-123.

[Do85] Donner, H.: *Normen schaffen Freiheit im Büro*, *com* Siemens-Magazine for: Computer & Communications, 4 (1985), pp. 8-13.

[EK89] Eirund,H. & Kreplin K.: *Knowledge Based Document Classification Supporting Integrated Docuemnt Handling*, in: Proceedings COIS, Palo Alto, CA (1989), pp 189-196.

[Fa85] Faloutsos, C., *Access Methods for Text,* Computing surveys, Vol. 17, No. 1, ACM (1985).

[FK85] Fikes, R., Kehler, T.: *The Role of Frame-based Representation in Reasoning,* in: Communications of the ACM, Vol. 28, No. 9 (1985), pp. 904-920.

[FWA85] Fox, M., Wright & J., Adam, D.: *Experience with SRL: an Analysis of a Frame-based Knowledge Representation,* Technical Report CMU-CS-81-135, Carnegie-Mellon University, Pittsburgh (1985.)

[HK87] Hull, R. & King, R.: *Semantic Database Modeling: Survey*, Applications, and Research Issues, in: ACM Computing Surveys, Vol. 19, No. 3 (987), pp. 201-260.

[HM78] Hammer & M., McLeod, D.: *The Semantic Data Model: A Modelling Mechanism for Data Base Applications,* in: Proc. of the ACM SIGMOD, Int. Conf. on Management of Data (1978), pp. 26-36.

[HBD90] Hönes, F., Bleisinger, R. & Dengel, A.: *Intelligent Word-based Text Recognition*, Proceedings OE-90, Symposium on Advances in Intelligent Systems (Machine Vision and System Integration), Boston, MA (1990).

[Ho85] Horak, W.: *Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization*, Computer (Okt. 1985,) pp 50-60.

[ISO85] International Standards Organization: *Information processing - text and office system - document structures - Part 2: Office Docuemnt Architecture,* 1985.

[JF88] Johnson & R. E., Foote, B.: *Designing Reusable Classes,* Journal of Object-Oriented Programming, June/July 1988, pp. 22-35.

[KBG89] Kim, W., Bertino, E. & Graza, J.F.: *Composite Objects Revisted*, in: Proc. of ACM SIGMOD Conf., Portland - Oregon (1989), pp.337-347.

[KR89] *The KBMS Prototype KRISYS* - User Manual, Version 2.3, Kaiserslautern, West Germany (1989.)

[Ma88a] Mattos, N.M.: *Abstraction Concepts: the Basis for Data and Knowledge Modeling,* in: 7th Int. Conf. on Entity-Relationship Approach, Rom, Italy (1988), pp. 331-350.

[Ma88b] Mattos, N.M.: *KRISYS - A Multi-Layered Prototype KBMS Supporting Knowledge Independence*, in: Proc. Int. Computer Science Conference - Artificial Intelligence: Theory and Application, Hong Kong (1988), pp. 31-38.

[MD91] N.M. Mattos & A. Dengel, *The Role of Abstraction Concepts and Their Built-In Reasoning in Document Representation and Structuring*, Proceedings Fourth Intl. Symposium on AI, Cancun, Mexico, November 1991

[MM89] Mattos, N.M. & Michels, M.: *Modeling with KRISYS: the Design Process of DB Applications Re-viewed*, in: Proc. the 8th Int. Conf. on Entity-Relationship Approach, Toronto, Canada (1989) pp..159-173.

[PM88] Peckham, J. & Maryanski, F.: *Sematic Data Models*, in: ACM Computing Surveys, Vol. 20, No. 3 (1988), pp. 153-189.

[RHMD87] Rosenthal, A., Heiler, S., Manola, F. & Dayal, U.: *Query Facilities for Part Hierarchies: Graph Traversal, Spatial Data, and Knowledge-Based Detail Supression*, Research Report, CCA, Cambridge, MA (1987).

[SB86] Stefik, M. & Bobrow, D.G.: *Object-Oriented Programming: Themes and Variations*, in: Al-Magazine, Vol. 6, No. 4, Winter 1986, pp. 40-62.

[SS77] Smith, J.M & Smith, D.C.P.: *Database Abstractions: Aggregation and Generalization*, in: ACM Transactions on Database Systems, Vol. 2, No. 2, June 1977, pp. 105-133.

[SM83] Salton, G. & McGill, M.J., *Introduction to Modern Information Retrieval*, McGraw Hill, 1983.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

# DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

# DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or per anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

**DFKI Research Reports**

**RR-92-43**
*Christoph Klauck, Jakob Mauss:* A Heuristic driven Parser for Attributed Node Labeled Graph Grammars and its Application to Feature Recognition in CIM
17 pages

**RR-92-44**
*Thomas Rist, Elisabeth André:* Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP
15 pages

**RR-92-45**
*Elisabeth André, Thomas Rist:* The Design of Illustrated Documents as a Planning Task
21 pages

**RR-92-46**
*Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster:* WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

**RR-92-47**
*Frank Bomarius:* A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

**RR-92-48**
*Bernhard Nebel, Jana Koehler:*
Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

**RR-92-49**
*Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:* Heuristic Classification for Automated CAPP
15 pages

**RR-92-50**
*Stephan Busemann:*
Generierung natürlicher Sprache
61 Seiten

**RR-92-51**
*Hans-Jürgen Bürckert, Werner Nutt:*
On Abduction and Answer Generation through Constrained Resolution
20 pages

**RR-92-52**
*Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul:* PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

**RR-92-53**
*Werner Stephan, Susanne Biundo:*
A New Logical Framework for Deductive Planning
15 pages

**RR-92-54**
*Harold Boley:* A Direkt Semantic Characterization of RELFUN
30 pages

**RR-92-55**
*John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen:* Natural Language Semantics and Compiler Technology
17 pages

**RR-92-56**
*Armin Laux:* Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

**RR-92-58**
*Franz Baader, Bernhard Hollunder:*
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

**RR-92-59**
*Karl Schlechta and David Makinson:* On Principles and Problems of Defeasible Inheritance
13 pages

**RR-92-60**
*Karl Schlechta:* Defaults, Preorder Semantics and Circumscription
19 pages

**RR-93-02**
*Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist:* Plan-based Integration of Natural Language and Graphics Generation
50 pages

**RR-93-03**
*Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi:* An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

**RR-93-04**
*Christoph Klauck, Johannes Schwagereit:* GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

**RR-93-05**
*Franz Baader, Klaus Schulz:* Combination Techniques and Decision Problems for Disunification
29 pages

**RR-93-06**
*Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux:* On Skolemization in Constrained Logics
40 pages

**RR-93-07**
*Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux:* Concept Logics with Function Symbols
36 pages

**RR-93-08**
*Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer:* COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

**RR-93-09**
*Philipp Hanschke, Jörg Würtz:* Satisfiability of the Smallest Binary Program
8 Seiten

**RR-93-10**
*Martin Buchheit, Francesco M. Donini, Andrea Schaerf:* Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

**RR-93-11**
*Bernhard Nebel, Hans-Juergen Buerckert:* Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

**RR-93-12**
*Pierre Sablayrolles:* A Two-Level Semantics for French Expressions of Motion
51 pages

**RR-93-13**
*Franz Baader, Karl Schlechta:* A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

**RR-93-14**
*Joachim Niehren, Andreas Podelski,Ralf Treinen:* Equational and Membership Constraints for Infinite Trees
33 pages

**RR-93-15**
*Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster:* PLUS - Plan-based User Support Final Project Report
33 pages

**RR-93-16**
*Gert Smolka, Martin Henz, Jörg Würtz:* Object-Oriented Concurrent Constraint Programming in Oz
17 pages

**RR-93-17**
*Rolf Backofen:* Regular Path Expressions in Feature Logic
37 pages

**RR-93-18**
*Klaus Schild:* Terminological Cycles and the Propositional $\mu$-Calculus
32 pages

**RR-93-20**
*Franz Baader, Bernhard Hollunder:* Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

**RR-93-22**
*Manfred Meyer, Jörg Müller:* Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

**RR-93-23**
*Andreas Dengel, Ottmar Lutzy:* Comparative Study of Connectionist Simulators
20 pages

**RR-93-24**
*Rainer Hoch, Andreas Dengel:* Document Highlighting — Message Classification in Printed Business Letters
17 pages

**RR-93-25**
*Klaus Fischer, Norbert Kuhn:* A DAI Approach to Modeling the Transportation Domain
93 pages

**RR-93-26**
*Jörg P. Müller, Markus Pischel:* The Agent Architecture InteRRaP: Concept and Application
99 pages

**RR-93-27**
*Hans-Ulrich Krieger:*
Derivation Without Lexical Rules
33 pages

**RR-93-28**
*Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker:* Feature-Based Allomorphy
8 pages

**RR-93-29**
*Armin Laux:* Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

**RR-93-33**
*Bernhard Nebel, Jana Koehler:*
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

**RR-93-34**
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

**RR-93-35**
*Harold Boley, François Bry, Ulrich Geske (Eds.):*
Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).

**RR-93-36**
*Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt:* Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

**RR-93-38**
*Stephan Baumann:* Document Recognition of Printed Scores and Transformation into MIDI
24 pages

**RR-93-40**
*Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:*
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

**RR-93-41**
*Winfried H. Graf:* LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

**RR-93-42**
*Hubert Comon, Ralf Treinen:*
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

**RR-93-45**
*Rainer Hoch:* On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

## DFKI Technical Memos

**TM-91-14**
*Rainer Bleisinger, Rainer Hoch, Andreas Dengel:*
ODA-based modeling for document analysis
14 pages

**TM-91-15**
*Stefan Busemann:* Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

**TM-92-01**
*Lijuan Zhang:* Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

**TM-92-02**
*Achim Schupeta:* Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

**TM-92-03**
*Mona Singh:*
A Cognitiv Analysis of Event Structure
21 pages

**TM-92-04**
*Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:*
On the Representation of Temporal Knowledge
61 pages

**TM-92-05**
*Franz Schmalhofer, Christoph Globig, Jörg Thoben:*
The refitting of plans by a human expert
10 pages

**TM-92-06**
*Otto Kühn, Franz Schmalhofer:* Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

**TM-92-08**
*Anne Kilger:* Realization of Tree Adjoining Grammars with Unification
27 pages

**TM-93-01**
*Otto Kühn, Andreas Birk:* Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

**TM-93-02**
*Pierre Sablayrolles, Achim Schupeta:*
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

**TM-93-03**
*Harold Boley, Ulrich Buhrmann, Christof Kremer:*
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

**DFKI Documents**

**D-92-19**
*Stefan Dittrich, Rainer Hoch:* Automatische, Deskriptor-basierte Unterstützung der Dokument-analyse zur Fokussierung und Klassifizierung von Geschäftsbriefen
107 Seiten

**D-92-21**
*Anne Schauder:* Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

**D-92-22**
*Werner Stein:* Indexing Principles for Relational Languages Applied to PROLOG Code Generation
80 pages

**D-92-23**
*Michael Herfert:* Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

**D-92-24**
*Jürgen Müller, Donald Steiner (Hrsg.):* Kooperierende Agenten
78 Seiten

**D-92-25**
*Martin Buchheit:* Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

**D-92-26**
*Enno Tolzmann:*
Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

**D-92-27**
*Martin Harm, Knut Hinkelmann, Thomas Labisch:* Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

**D-92-28**
*Klaus-Peter Gores, Rainer Bleisinger:* Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

**D-93-01**
*Philipp Hanschke, Thom Frühwirth:* Terminological Reasoning with Constraint Handling Rules
12 pages

**D-93-02**
*Gabriele Schmidt, Frank Peters, Gernod Laufkötter:* User Manual of COKAM+
23 pages

**D-93-03**
*Stephan Busemann, Karin Harbusch(Eds.):*
DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

**D-93-04**
DFKI Wissenschaftlich-Technischer Jahresbericht 1992
194 Seiten

**D-93-05**
*Elisabeth André, Winfried Graf, Jochen Heinsohn, Bernhard Nebel, Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahlster:*
PPP: Personalized Plan-Based Presenter
70 pages

**D-93-06**
*Jürgen Müller (Hrsg.):*
Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz Saarbrücken 29.-30. April 1993
235 Seiten
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).

**D-93-07**
*Klaus-Peter Gores, Rainer Bleisinger:*
Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse
53 Seiten

**D-93-08**
*Thomas Kieninger, Rainer Hoch:* Ein Generator mit Anfragesystem für strukturierte Wörterbücher zur Unterstützung von Texterkennung und Textanalyse
125 Seiten

**D-93-09**
*Hans-Ulrich Krieger, Ulrich Schäfer:*
TDL ExtraLight User's Guide
35 pages

**D-93-10**
*Elizabeth Hinkelman, Markus Vonerden,Christoph Jung:* Natural Language Software Registry
(Second Edition)
174 pages

**D-93-11**
*Knut Hinkelmann, Armin Laux (Eds.):*
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

**D-93-12**
*Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:*
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

**D-93-14**
*Manfred Meyer (Ed.):* Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).