ON CONSISTENCY OF INFORMATION MODELS

BENGT LUNDBERG

Abstract.
During the last years a number of approaches to information modeling have been presented. An information model is then assumed to be expressed in some formalism based on a set of basic concepts and construction rules. Some approaches also include inference rules, but few include consistency criteria for information models. Two different approaches to information modeling have been analyzed within the framework of first-order predicate logic. In particular, their consistency criteria are compared with that of predicate logic. The approaches are completely expressible in predicate logic and the consistency criteria have a logical counterpart only when a set of implicit assumptions is stated explicitly.

1. Introduction.
During the last years a number of approaches to information modeling have been presented, usually including a set of basic concepts and a formalism for information models. Sometimes also a method for information modeling is suggested.

The basic concepts employed are objects (entities) and associations between objects, classified in object classes and association classes. An information model is seen as a formal model of the abstract knowledge about an application and it is intended to be true in the possible states of the application. The formalisms employed are usually intended to reflect application oriented concepts and the semantics of the relationships between them. While the formalisms as such are formally defined, most approaches to information modeling have not defined inference rules for their formalisms. In particular, consistency criteria for information models are almost never defined.

In this report the consistency criteria of two approaches, the DBDA approach [14] and the relation model (as proposed in [9]), are analyzed within the framework of (first-order) predicate logic (cf. Ch. 3). In order to perform this task the approaches are first analyzed and their predicate logic correspondences are established (cf. Ch. 2). The main results of this analysis is that information models expressed in the formalism of the approaches above can be completely described in the formalism of predicate logic and that the proposed consistency criteria have a logical counterpart only when a set of implicit assumptions are stated explicitly.

This work is supported by the National Swedish Board for Technical Development.
Received May 9, 1980. Revised Jan. 22, 1981.
2. First-order predicate equivalence of information models.

2.1 The framework for analysis.

Two approaches to information modeling will be analyzed in the framework of (first-order) predicate logic. The employed logical system is a standard one, e.g. as defined in [10], with the following extensions:

- a logical system with equality is assumed. I.e. the binary predicate "=" is employed as well as the axioms of equality.
- the binary predicate "e" denoting membership is assumed.

2.2 The Data Base Design Aid (DBDA).

DBDA is basically a tool for practical data base design and as such employing a set of basic concepts and a formalism. Further, DBDA is a typical example of a family of similar approaches which are based on binary associations between objects, cf. e.g. [16, 12, 1].

The basic concepts in DBDA are object classes (called data elements) and association classes (called associations). An object class is constituted by a set of "similar" objects, i.e. the objects satisfying the same predicate. Consider the object class EMP, which is assumed to denote the class of employees of an application. The object class EMP can then be defined in predicate logic in the following way [13]:

\[ \forall x (x \in \text{EMP} \leftrightarrow \text{employee}(x)) . \]

An association class is constituted by the set of associations satisfying a binary predicate, e.g.

\[ \forall x \forall y \langle x, y \rangle \in \text{HAS-SALARY} \leftrightarrow \text{has-salary}(x, y) \]

In DBDA an association class is represented (declared) as follows, e.g.

(EMP, SAL)

where the symbols EMP and SAL denote the object classes to which the objects of the associations belong, i.e. it corresponds to

\[ \forall x \forall y \text{has-salary}(x, y) \rightarrow \text{employee}(x) \& \text{salary}(y) . \]

Statements of this kind will here be referred to as domain declarations.

In DBDA the association classes in general do not have names. This implies that they have to be identified by the names of the object classes and that only one association class may hold between two object classes. Optionally, association classes may be given names (labels). However, the impact of such names on information modeling is not discussed in the report [14].

Three types of association classes are identified in DBDA:

(1) simple associations (type 1). If the association class above is of type 1 it is represented as

(EMP, SAL) = 1.