

91-55

ET

Faculteit der Economische Wetenschappen en Econometrie

05348

Inherently Flexible Information Systems

M. Boogaard
M.J.J.A. Gambin
E.R.K. Spoor and
R.J. Veldwijk

Research Memorandum 1991-55



Inherently Flexible Information Systems

M BOOGAARD, M J J A GAMBIN, E R K SPOOR, and R J VELDWIJK

Vrije Universiteit
Faculteit der Economische Wetenschappen en Econometrie
Vakgroep Bestuurlijke Informatiekunde
De Boelelaan 1105
1081 HV AMSTERDAM

Evolving information requirements often force the restructuring of database structures, resulting in major adjustments in many application programs that run against the database. The introduction of inherent flexibility by means of an integrated data dictionary provides a way to reduce maintenance effort significantly. The development of models that explicitly support the construction of inherently flexible information systems is the overall objective of the MESDAG Research Group. This paper describes one of the directions within this research framework.

1. INTRODUCTION

Formerly, the acceleration of the usage of information technology by organizations was primarily connected with the advances in hardware technology. However, nowadays software has become the leading component of computerization. The shift of dominance from hardware to software technology is caused by the awareness of the rapidly growing gap between the supply and demand of information systems, i.e., the so-called application backlog or software crisis.

Another aspect pertaining the current information technology is the automation of processes in unstable and unstructured environments. These "new" environments, for example management information systems and decision support systems, are characterized by different object structures and highly evolving information requirements. The continually evolving information requirements of organizations are intrinsic to the nature of information system utilization (see Lehman 1985) and organizational processes.

Both features of the contemporary information technology described, result from the lack of inherent flexibility of the information systems which is a predominant contributor to the current maintenance mess (see Martin and McClure 1985). Traditional modelling methodologies and tools are not equipped for explicit support of the development of flexible information systems. This observation forms the basis of the mission of the MESDAG Research Group¹, i.e., to prove the feasibility of developing inherently flexible information systems by introducing higher levels of logical data independence. The MESDAG philosophy stems from the premise that *inherent flexibility requires self-knowledge* (see Hofstadter 1979).

1

The MESDAG project is a joint project endorsed by three organizations in the Netherlands: Netherlands Railways Company, RAET N.V., and the Vrije Universiteit of Amsterdam. MESDAG is the acronym of MEta Systems Design And Generation. The MESDAG Research Group consists of the following four members: dr. E.R.K. Spoor (associate professor at the Vrije Universiteit), drs. R.J. Veldwijk (consultant at RAET N.V.), M.J.J.A. Gambin (researcher at the Vrije Universiteit), and drs. M. Boogaard (assistant researcher at the Vrije Universiteit).

The core of self-knowledge consists of simple rules, metarules to change these simple rules, metametarules to changes the metarules, etc. Consequently, flexibility proceeds from an enormous amount of rules on different levels. Information systems model the rules of their universe of discourse. To achieve flexibility, the information systems thus should contain a metamodel, a metametamodel, etc. The uncontrollable accumulation of metamodels can be eliminated by storing the description of the metamodel in the metamodel itself (see Ross 1981, Nijssen and Halpin 1989, and Veldwijk et al. 1991c). This leads to a self-referential metamodel that replaces the complete hierarchy of metamodels. Integration of this metamodel with the "simple model" blurs the distinction between the different levels of both models which results in inherently flexible information systems (see Veldwijk et al. 1991a-c, Spoor et al. 1991, and Boogaard et al. 1991).

This paper elaborates on one of the objectives of the MESDAG Research Group. It is based on the research plan of the first author. It focuses on an elementary operations approach aimed at the automatic restructuring of information systems. Section 2 describes the "state of the art" with respect to maintenance and the extent of the flexibility of current information systems. Furthermore, this section justifies the selection of the relational model as the starting-point. Section 3 introduces the research approximation to be followed. The first subsection of section 3 illuminates the concepts of an automatic restructuring mechanism based on the MESDAG philosophy. The second subsection explores the application of this restructuring mechanism to the creation of a data independent object representation which introduces flexibility for end-users. Section 4 concludes the paper by listing suggestions for future research directions and other applications of the concepts presented.

2. STATE OF THE ART

There exist several designing actions that can ease the maintenance of information systems. For example, (1) adapt maintainability as a major principle of information system design, (2) minimize the complexity of the information system and application program structures, (3) use a structured development approach, (4) increase productivity, etc. by using new technologies like CASE, 4GL, and RDBMSs. However, these solutions address only the symptoms of the underlying origin of maintenance, i.e., the lack of inherent flexibility of information systems (see Boogaard et al. 1991). In conclusion, current software technologies mainly focus on productivity, and technical and development facets rather than flexibility.

Object oriented approaches (see, e.g., Meyer 1988) and the relational model (see, e.g., Codd 1990) are concerned with the development of flexible information systems. The premise of the MESDAG Research Group is the relational model, because the object oriented approach, although conceptually promising, is not fully crystallized yet. Consequently, there is no agreement on *the* object oriented model and a formal definition of the approach is not yet at hand. This deficiency restricts the development of a self-referential metamodel of the object oriented model which is essential to accomplish inherently flexible information systems. Furthermore, the concepts to be presented can easily be implemented in RDBMS environments which are widely used nowadays.

The relational model, however, is a formal, mathematically based data model, which can be modelled using the relational model itself (see Veldwijk et al. 1991c). It claims to introduce a degree of flexibility by means of physical and logical data independence (see Figure 1).

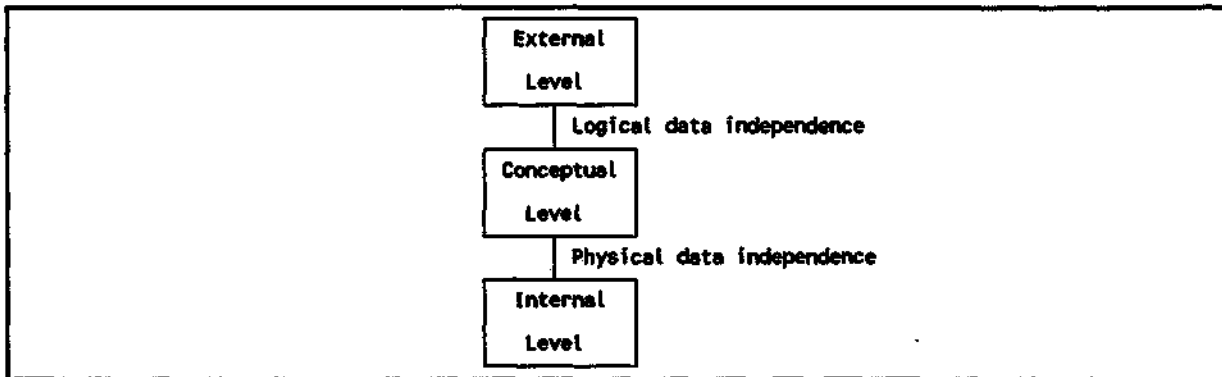


Figure 1. Data independence

Although current RDBMSs (Relational Database Management Systems) provide physical data independence to a large extent, the level of logical data independence is still limited. The relational model and RDBMSs endeavour to achieve logical data independence using the view concept (see Figure 2).

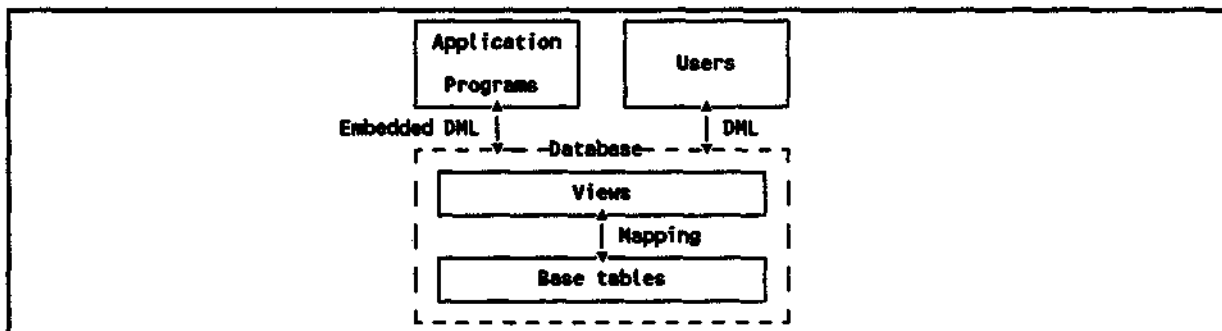


Figure 2. Current situation

However, the use of views results in a unsatisfactory level of logical data independence, even when RDBMSs adequately support view updatability (see Veldwijk 1991a). In conclusion, the current level of logical data independence attained does not result in inherently flexible information systems. However, as will be explained in the subsequent sections, the relational model can be used to accomplish these systems because a self-referential metamodel of the relational model can be developed².

²

Actually, the internal catalog of a RDBMS is a metamodel of the relational model. However, the catalogs of contemporary RDBMSs do not fully support all features of the relational model (see Codd 1990). Furthermore, these catalogs cannot be used integrated with the database structures because the catalogs are masked (except for retrieval) and can only be altered indirectly by means of DDL (Data Definition Language).

3. MODUS OPERANDI

The lack of logical data independence has severe negative repercussions on both application programs and users. This section describes the modus operandi to be followed striving after logical data independence to both of them. It will be amplified by means of the following two steps:

1. Logical data independence from an application program's point of view.
2. Logical data independence from a user's point of view.

During these steps an active, self-referential data dictionary will be used. This data dictionary is the metamodel of the relational model structured by the relational model itself and is integrated with the database structure.

3.1 Step 1: Application Programs

The first step concentrates on the introduction of logical data independence for application programs. Figure 3 illustrates the underlying architecture for this step.

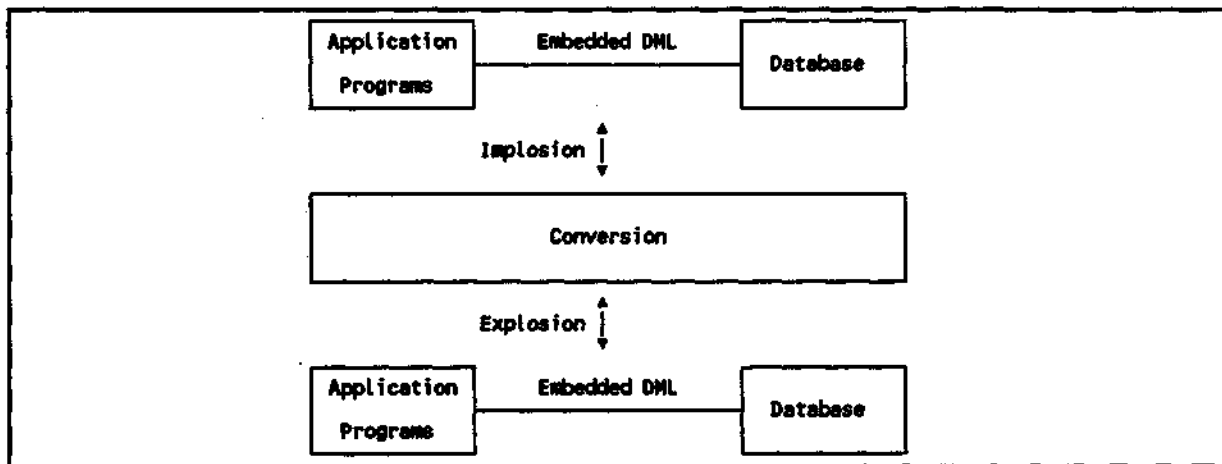


Figure 3. Application program's point of view

Whenever a database structure is no longer in the condition to represent the status of its universe of discourse, alteration of the database structure is necessary. Part of these changes are *information-preserving* and can be considered restructuring.

For example, consider the following database structure (the primary key of the relations is underlined and the characters represent the aliases designated to the columns which is necessary to identify the columns during the EDSO procedure to be explained, see Figure 4 below):

DEPARTMENT	<u>(DEPNQ,</u> DEPNAME, BUDGET)
	A B C
EMPLOYEE	<u>(EMPNO,</u> EMPNAME, DEPNO)
	D E F

In the original situation, the database structure reflects a one-to-many relationship between DEPARTMENT and EMPLOYEE which means that a department can employ more than one employee but an employee can only work on one department. Suppose that in consequence of an alteration in the environment, the relationship between DEPARTMENT and EMPLOYEE must change into a many-to-many relationship because an employee can now work on more than one department. The following database structure meets this requirement:

DEPARTMENT	(<u>DEPNO</u> , DEPNAME, BUDGET)
	A B C
ASSIGNMENT	(<u>DEPNO</u> , <u>EMPNO</u>)
	F G
EMPLOYEE	(<u>EMPNO</u> , EMPNAME)
	D E

In principle, restructuring processes must be analysed separately. However, it is possible to decompose the modifications required into a certain number of elementary operations, i.e., Elementary Database Structure Operations (EDSOs). These operations are generalized and thus applicable for every database structure.

Whenever the EDSOs required are determined, the procedure consists of three steps that must be executed for each EDSO selected:

1. Implosion: each relation of the source database and all the embedded DML (Data Manipulation Language) queries of the application programs affected must be imploded into the data dictionary;
2. Conversion: having checked several constraints, both the imploded relations and DML queries are converted conform to the specifications of the structure modification, which is specific for the EDSO executed;
3. Explosion: the original data content is re-stored into the new database structure, and the restructured queries are embedded into their original application programs.

Figure 4 illustrates the working of the EDSO required for the example, i.e., the transformation of a one-to-many to a many-to-many relationship. It also shows the consequences for an exemplary embedded query which is formulated in SQL (Structured Query Language). It should be noted that the scope of the paper is to prove the feasibility of the EDSO procedure rather than to describe it in detail (see Veldwijk 1991a for an in-depth analysis). Consequently, only a concise description of the procedure is given. Furthermore, the EDSO procedure described is a conceptual framework. It is conceivable to implement it in a different way using this framework.

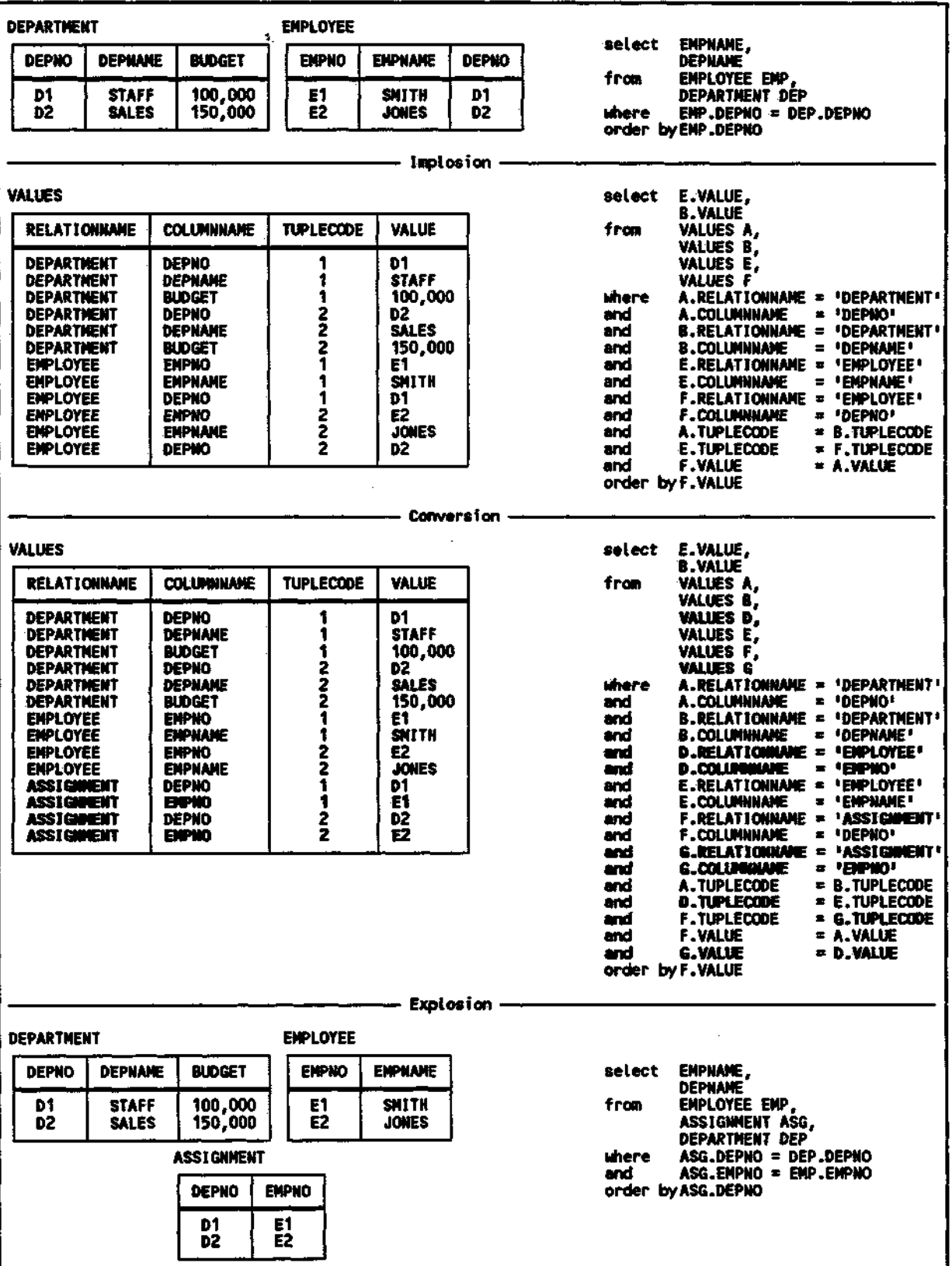


Figure 4. EDSO procedure

Every database structure can be transferred to the content of the VALUES relation³. Implosion means that each value of a relation together with its structural characteristics is incorporated in the data dictionary relation VALUES. Hence, the distinction between structural data (metadata) and "simple" data disappears. The only structural description that is left, is the invariant structure of VALUES:

VALUES (RELATIONNAME, COLUMNNAME, TUPLECODE, VALUE)

Except for TUPLECODE, the information represented by the columns of VALUES was explicitly present in the initial situation. The addition of TUPLECODE is necessary in order to reflect the implicit connection between the values of the original tuples. However, TUPLECODE is not added to order the tuples and thus does not violate relational principles.

The DML queries affected are transformed conform to the implosion of the original database structure. These queries are also stored in data dictionary relations.

Consequently, every alteration of the database structure and as a result the DML queries, now requires modification of the content of data dictionary relations. This can be directly accomplished by means of DML. Normally, DDL (Data Definition Language) would be needed to change the source database structure into the required database structure. Furthermore, several complex operations would be required to adjust both the content of the source relations and the embedded DML queries of application programs to the new database structure.

The determination of a powerful or even complete set of EDSOs will be crucial in this step. With this set of EDSOs every desired database alteration and application program adjustment can be executed automatically on the basis of one or more EDSOs. Other conceivable EDSOs are, for instance:

- The transformation of a many-to-many into a one-to-many relationship, i.e., the inversion of the EDSO described.
- The transfer of a descriptive column of a relation on the one-side of a relationship to the relation on the many-side.
- The transfer of a descriptive column of a relation on the many-side of a relationship to the relation on the one-side.

The following important consideration must be taken into account. The application programs still depend on the logical structure of the database after the restructuring. However, this dependence can be *bypassed* using the automatic restructuring tool. Such a restructuring tool supports the database administrator during the maintenance process and permits enhancements of the database structure, which would probably be denied in the original situation. Thus, a surrogate logical data independence can be achieved from an application program's point of view.

3.2 Step 2: Users

The second step involves the user's point of view. The problem of logical data dependence appears when users try to translate their requests into DML queries. The query contains several database structural terms and thus depends on the current structure of the database. Consequently, users must have a complete overview of the database structure at that point of time. As a result, a future alteration of the database structure affects the query composed. The original information request, however, remains the same.

³

Because the VALUES relation is a canonical form, even the metametadate, i.e., the data dictionary description itself, can be stored in the VALUES relation.

Two important conclusions of the first step form the foundation of the second step towards a higher level of logical data independence. First, the utilization of DML will always lead to a structural dependence even when the database structure is invariant. The exemplary queries of Figure 4 on the VALUES relation still contain structural information, i.e., the equations with RELATIONNAMEs, COLUMNNAMEs, and TUPLECODEs in the WHERE-clauses. Second, the EDSO concept is a useful vehicle to alter existing structures automatically. Figure 5 shows the conceptual approach which enables logical data independence from a user's point of view.

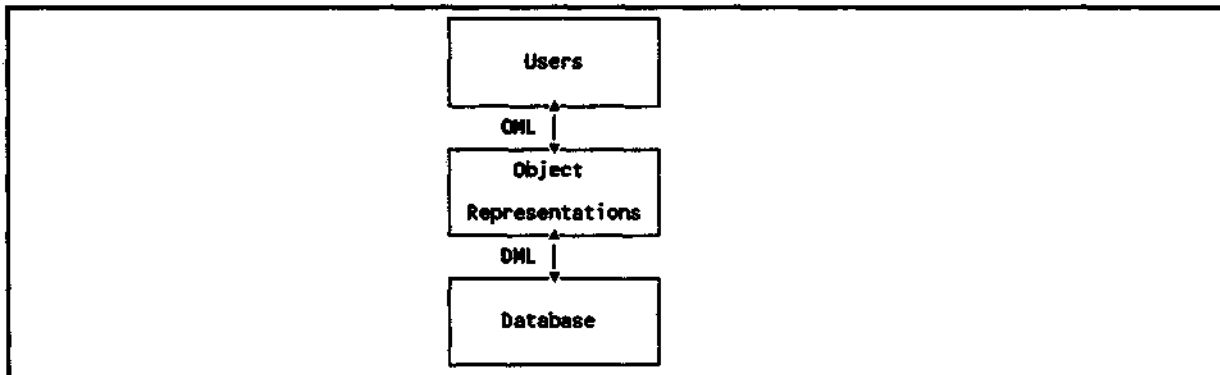


Figure 5. User's point of view

Instead of interacting directly with the database structure, each user can define his own perspective in terms of objects, properties, and relationships between objects, i.e., an object representation.

Consequently, the users no longer depend on the logical structure of the underlying database. Thus, the users thus do *not* have to abide by a discipline in contrast with Codd's allegation (1990, p. 5). The database structure (using EDSOs) may change without impairing the perspectives of the users, i.e., *EDSO transparency*. Furthermore, the object representation can be altered by users (whenever he or she wants) within a framework of rules without affecting the database structure by means of the execution of conceptual EDSOs on his or her object representation.

As far as the users are concerned, the object representations can be considered external schemata. Thus, from a user's point of view logical data independence is achieved as originally described by means of the three schema distinction (see Figure 1). An important observation is that the users are now dependent on their object representation. This seems just a postponement of the problem. However, the users can change their object representations whenever and in any information-preserving way they want without affecting other users and application programs. Furthermore, the structure of the database can change without affecting the perceptions of the users in the first instance. The users interact with the database using an object manipulation language (OML) which contains more natural constructs than DML. Consequently, OML simplifies the interaction between users and the stored data considerably.

4. CONCLUDING REMARKS

In this paper one of the objectives of the MESDAG Research Group is explored. The MESDAG philosophy is applied to the maintenance of information systems within the context of the relational model. Furthermore, it is stated that the concepts for maintenance also form the basis for database structure-independent perspectives of end-users. However, the applicability of the underlying philosophy is not limited to automatic restructuring and users' database inquiry only. Without listing a complete set of possibilities, these prospects are illustrated by two directions for further research which are directly related to the content of this paper.

The necessity of language development (OML) requires further research and can be seen as a bottom-up approach to accomplish the user's interaction with information systems on the basis of natural language. If an OML can be formally defined, it is also applicable for the application programs, i.e., embedded OML. Figure 6 depicts the resulting ideal situation of logical data independence.

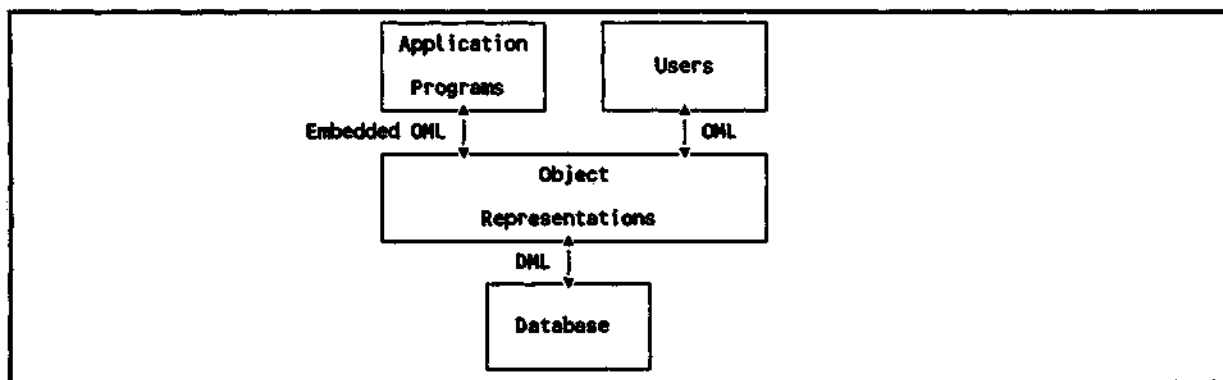


Figure 6. Further research

In conclusion, logical data independence can be achieved as meant by the three level distinction, i.e., internal, conceptual and external schema (see Figure 1). The object representations conform to the definition of external schemata for both application programs and users. Comparing this to the current situation (see Figure 2), "views" on the database are transferred before instead of after the use of DML.

Obviously, the presented approach affects the development process of information systems in highly dynamic environments. However, the merits of the utilization of the underlying data dictionary not only cover these environments. The data dictionary can also be used passively to control and support most aspects of the development process. Furthermore, it is possible to use the data dictionary partly actively, for example to support structures in which objects belong to multiple object types, e.g., generalization (see Boogaard 1991). The decision on how to use the data dictionary, or in other words how flexible should the information system be, should be an essential element of the first stage of the development process. Moreover, a methodology must be constructed to measure the flexibility required for the information system to be developed. In conclusion, application of the MESDAG philosophy should result in adjustments of the current information system development methodologies.

REFERENCES

- Boogaard, M., Veldwijk, R.J., Spoor, E.R.K. and Dijk, M.V. van (1991) "On Generalization in the Relational Model," *Serie Research Memoranda*, Research Memorandum 1991-37, Amsterdam: Faculteit der Economische Wetenschappen en Econometrie, Vrije Universiteit Amsterdam (submitted to IFIP WG 8.1 Working Conference on "Information Systems Concepts: Improving the Understanding", Alexandria/Egypt, April 1992).
- Buitendijk, R.B. (1991), *Towards an Effective Use of Relational Database Management Systems*, Amsterdam: Thesis Publishers (no 4. of the Tinbergen Institute research Series).
- Codd, E.F. (1990), *The Relational Model for Database Management: Version 2*, Reading, Massachusetts: Addison-Wesley Publishing Company.
- Lehman, M.M. and Belady, L.A. (1985), *Program Evolution: Processes of Software Change*, London: Academic Press.
- Martin, J. and McClure, C. (1985), "The Maintenance Mess," *Software World*, 14, 4, pp. 10-16.
- Meyer, B. (1988), *Object-Oriented Software Construction*, New York: Prentice Hall.
- Nijssen, G.M. and Halpin, T.A. (1989), *Conceptual Schema and Relational Database Design: A Fact Oriented Approach*, New York: Prentice Hall.
- Ross, R.G. (1981), *Data Dictionaries and Data Administration: Concepts and Practices for Data Resource Management*, New York: AMACOM.
- Spoor, E.R.K., Veldwijk, R.J., and Boogaard, M. (1991), "The Transformation of Extended Entity Relationship Generalization Hierarchies into Tables and Meta Tables," *Serie Research Memoranda*, Research Memorandum 1991-47, Amsterdam: Faculteit der Economische Wetenschappen en Econometrie, Vrije Universiteit Amsterdam (submitted to the European Journal of Information Systems).
- Veldwijk, R.J., Boogaard, M., Dijk, van M.V., and Spoor, E.R.K. (1991a), "EDSOs, Implosion and Explosion: Concepts to Automate Part of Application Maintenance of Relational Databases," *Information and Software Technology*, 33, 5, pp. 343-350.
- Veldwijk, R.J., Spoor, E.R.K., Boogaard, M., and Dijk, M.V. van (1991b), "On the Expressive Power of the Relational Model: A Database Designer's Point of View," *Serie Research Memoranda*, Research Memorandum 1991-49, Amsterdam: Faculteit der Economische Wetenschappen en Econometrie, Vrije Universiteit Amsterdam (to appear in the Proceedings of the Twelfth Annual International Conference on Information Systems, New York, December 1991).
- Veldwijk, R.J., Buitendijk, R.B., Spoor, E.R.K., and Boogaard, M. (1991c), "Towards a Catalog Standard for the Relational Model Version 2: A Manifesto," *Serie Research Memoranda*, Research Memorandum 1991-50, Amsterdam: Faculteit der Economische Wetenschappen en Econometrie, Vrije Universiteit Amsterdam.