



Doctoral Thesis

## Wissensbasierte Werkzeuge für den Datenbank-Entwurf

**Author(s):**

Brägger, Richard P.

**Publication Date:**

1987

**Permanent Link:**

<https://doi.org/10.3929/ethz-a-000400698> →

**Rights / License:**

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

*Wissensbasierte Werkzeuge für den  
Datenbank-Entwurf*

ABHANDLUNG

zur Erlangung des Titels eines  
Doktors der Technischen Wissenschaften  
der  
EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE (ETH)  
ZÜRICH

vorgelegt von  
RICHARD PAUL BRÄGGER  
dipl. Math. ETH  
geboren am 31. Dezember 1956  
von Ebnet-Kappel

Angenommen auf Antrag von  
Prof. Dr. C.A. Zehnder, Referent  
Prof. Dr. H.-J. Appelrath, Korreferent  
Dr. R. Marti, Korreferent

## Zusammenfassung

Beim *Entwurf von Datenschemata* für Datenbank-Anwendungen ist ein ingenieurmässiges Vorgehen notwendig, um zu Lösungen zu gelangen, die vorgegebenen Qualitätsansprüchen genügen. Unbedingte Voraussetzung dafür sind *Werkzeuge*, die den Modellierungsvorgang auf allen Ebenen und auf unterschiedliche Arten unterstützen. Die vorliegende Arbeit befasst sich mit dem Entwickeln solcher Werkzeuge und den damit verbundenen Problemen.

Als Resultat wird das Entwurfssystem Prometheus vorgestellt, das den konzeptionellen Datenbank-Entwurf unterstützt. Es basiert auf einem erweiterten *Entitäten-Beziehungsmodell*. Alle Teilschritte sind separat durchführbar, es sind temporäre Inkonsistenzen und Definitionslücken zugelassen. Die Konsequenzen jedes Entwurfsschrittes werden aufgezeigt, zudem bestehen mächtige Operationen, die jederzeit beliebige Änderungen des Datenbank-Schemas ermöglichen.

Wesentlicher Bestandteil der vorgestellten Entwurfsmethode, das konzeptionelle Prototyping, ist die *Prototyp-Erprobung*. Nebst den Datenstrukturen und den zusätzlichen Konsistenzbedingungen kann der Entwerfer auch die elementaren Änderungstransaktionen modellieren und anschliessend mit einer Prototyp-Datenbank austesten. Fehler und Mängel im Entwurf sind so frühzeitig erkennbar. Die Benutzerschnittstelle des Werkzeuges basiert auf einer *graphischen Darstellung* des Entitäten-Blockdiagramms, welches für die interaktive Ein- und Ausgabe verwendet wird.

Spezielles Gewicht wird auf die Darstellung des internen *Wissens des Entwurfswerkzeuges* gelegt. Ein Vergleich von Wissensdarstellungsformen in Expertensystemen und Datenmodellen zeigt auf, dass Konsistenzbedingungen und Produktionsregeln integriert werden können. Daraus lassen sich zahlreiche Folgerungen für Experten-Datenbanksysteme ziehen. Durch die Verwendung einer *Metadatenbank*, in der gleichzeitig die definierte Datenbankbeschreibung und das Systemwissen verwaltet werden, lässt sich das Entwurfswerkzeug als Experten-Datenbank-Anwendung realisieren. Interessant ist die Erkenntnis, dass das Entwurfswissen nichts anderes als die Konsistenzbedingungen des Metaschemas sind. Daher lässt sich das Systemwissen mit dem gleichen Entwurfswerkzeug entwickeln und stetig erweitern.

Für Expertensysteme existiert noch keine vertiefte Entwurfsmethodik. Die in dieser Arbeit vorgestellte Methode ist dann von Interesse, wenn permanente Daten in solchen Systemen verwendet werden sollen. Die Unterscheidung in konstitutive und regulative Konsistenzbedingungen ermöglicht Optimierungen auf der physischen Ebene und eine verbesserte Verwaltung der zahlreichen Regeln.

## *Abstract*

The *design of the data schema* is a very important step in the development of a database application. An engineering approach is needed for developing concepts that fit the given quality requirements. Such an approach must be supported by *tools* which assist the designer during all steps of the design process and support the modeling of the schema on various abstraction levels. This thesis is concerned with database design tools that fulfill the various requirements.

As a main result of the thesis the system Prometheus is presented. It is a database design tool that supports an interactive development of the conceptual schema, based on an extended *Entity-Relationship-Model*. All design steps can be executed independently, Prometheus allows temporary inconsistencies and unknown data, but also supports its elimination. The consequence of a design decision is made visible to the user immediately. To be flexible the tool offers all kind of restructuring functions.

An important part of the presented design methodology, the Conceptual-Prototyping, is the *testing of a prototype database*. The user of Prometheus not only defines the data structures and additional semantic integrity constraints but also models the elementary update transactions. Afterwards it is possible to test the prototype database by executing these transactions. As a result, errors and shortcomings in the data schema can be detected in an early stage of the design process. The user interface of the tool is based on a *graphical representation* of the entity-block-diagram. It is used for interactive input and output.

The architecture of the design tool is strongly influenced by the representation of the system's knowledge. A comparison of knowledge representations used in expert systems and data models shows that semantic integrity constraints and production rules can be integrated. Based on this similarity several conclusions for expert database systems can be drawn. By using a meta database where the description of the developer's database and the system's knowledge is stored together, the design tool can be implemented as expert database application. It is shown that the design knowledge in fact consists of the semantic integrity constraints on the meta database level. Therefore, the knowledge of the design tool can be managed by the tool itself.

For the development of expert systems, an exact and generally accepted methodology is not yet known. The design method presented in this thesis can be used provided that permanent data is going to be used in an expert system. The distinction between different kinds of integrity constraints allows optimizations on the physical level and an improved management of all kinds of rules.