The apparent ease with which Entity-Relationship (ER) models can be understood and created can lead to false impressions of the quality of those models, which can result in non-functional dependencies in a relational schema. Such problems should be resolved at the ER model stage rather than by subsequent normalization.

We have developed an approach to identifying potential redundancy and ambiguity within an ER model, and implemented our approach in a prototype CASE tool, ERD2. ERD2 interacts with the modeller to help resolve these problems to improve the quality of the ER model and, hence, of the subsequently derived database schema.

The approach is generally applicable to a range of graphical modelling notations that utilise the same basic constructs.
The Trouble with Graphical Languages...

... is that you can draw silly things!

Diagrams such as Entity Relationship (ER) Models are intended for communication between analyst and client. Although they seem to be easy to understand, there is evidence that even analysts may not understand them completely.

Furthermore, interaction with clients tends to be confined to the early stages of analysis, whereas questions of potential redundancy are often not resolved until later, perhaps without full knowledge of the client domain.

One problem can be the inclusion in the ER model of relationships which appear to be specified in an initial system description, but which are, in fact, implied by other parts of the model. Such relationships are redundant, and, unless they are detected and removed, they will lead to redundancy in any relational schema synthesised from the model.

A System Description

A contracting company engages temporary (contract) employees to work solely on a single project, which is sponsored by a single client. The project is managed by one of the company’s permanent staff. Every contract employee, who is identified within the company by the role they play in his or her project, is paid for in full by the client of the project on which (s)he is working, and the contract employee reports to a manager who is a member of the company’s permanent staff. Contract employees, Managers and Clients all have names; Client names are unique. Projects are identified by their title, and Managers have a unique (company) identifier. Client addresses and the start date of each project are also recorded.

Researchers

David S. Bowers - Department of Computing, The Open University

Recent Publications


Collaborators: Alex Gray, Professor of Advanced Information Systems, Cardiff University
A relationship between two entities has four components: the degree (at each end), and the optionality (participation of each entity in relationship).

The signature of a relationship, in each direction, is either function or multifunction, and either total or partial.

The composition of relationship signatures shown here can be generalised for all relationship signatures.

Consider the four parts of a relationship, and code each part:

1 if single valued (function)
1 if mandatory (total)
0 otherwise

The overall signature of a composite relationship is the “AND” of the signature codes of the component relationships.

It is trivial to deduce these equivalent signatures for each of the 16 compositions of many:one relationships.

The composition is recursive.
The ERD2 Tool in Action

An Initial Entity Relationship Model

The Tool’s Interactions with the Modeller

The “corrected” Model

Researchers
David S. Bowers - Department of Computing, The Open University
Kostas Vassiliadis, Andrea Levati, Mark Stuart - Final Year project students, University of Surrey

Recent Publications