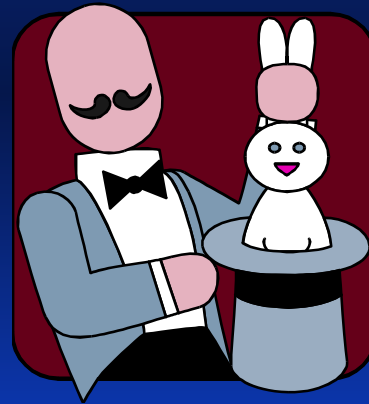


*Oracle Designer:
The Magic of Information Flow*



Paper #323

Peter Koletzke, Millennia Vision Corp.

Douglas Scherer, Core Paradigm

It Seems Like Magic

You put information in the repository

It goes somewhere

You get code or reports out

But it's not what you expect

*Kind of like pulling a rabbit out of your
hat*

It Should Not Seem Like Magic

Knowing about how information flows is key to successful work in Designer

Helps you discriminate what properties are important

Prepare for generation

Minimize errors

- Errors, inconsistencies, missing data*
- Errors in early stages can be expensive later*

Agenda

Examine categories of information flow

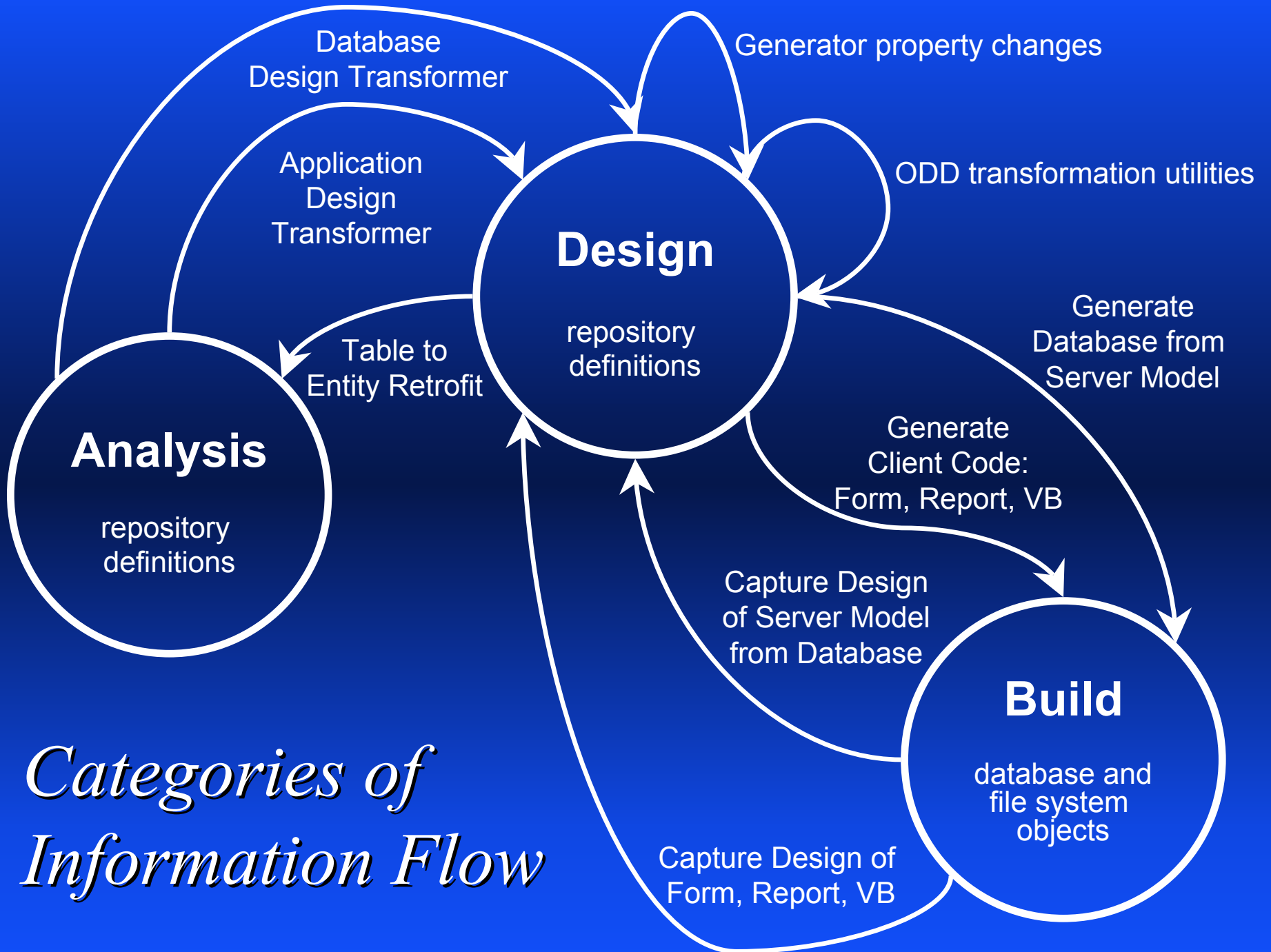
- Some in more detail than others*

List effects of specific property flows

Demonstrate some of the transformations

Small subset of property flows

Concentrate on flows within the repository only



Categories of Information Flow

Categories of Information Flow

Analysis to Design

Design to Build (Generation)

Build to Design

- *From the database*
- *From application modules*

Design to Analysis

Design to Design

What About Other Methodologies?

RAD

Bottom-up

Middle-out

You may not use all the transformers

*You still need to know what properties
affect what other properties*

Analysis to Design

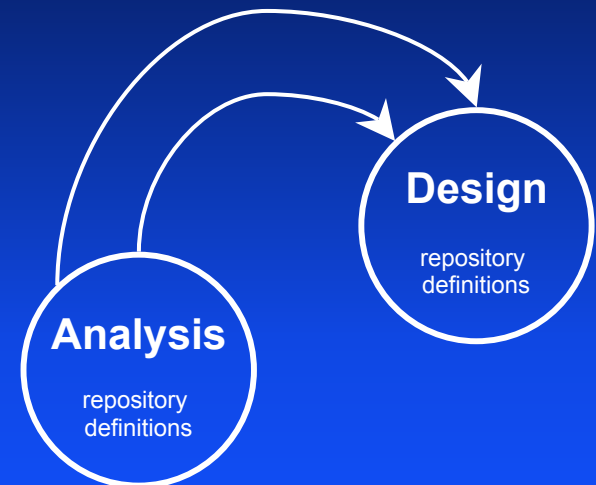
Logical to physical

- *Data: tables, columns, FK columns, constraints*
- *Process: Forms, Reports, manual modules*

Database Design Transformer (DDT)

- *Make changes to entities before opening DDT*
- *Start the DDT from RON (more in demo)*

Application Design Transformer (ADT)



DEMO

Squash your bugs with DDT

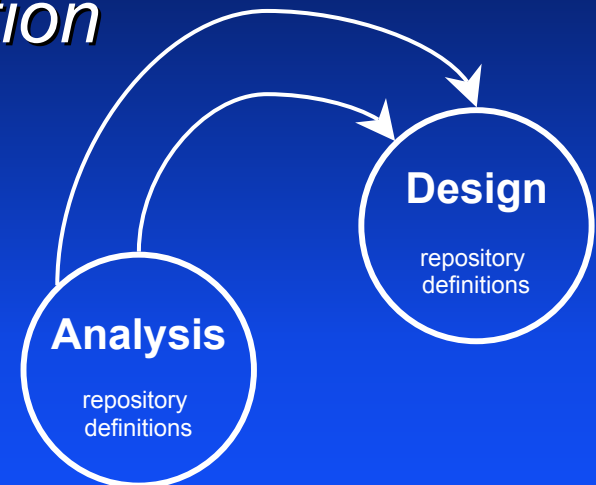
Analysis to Design

Entity → Table

- *Plural → Name & Display Title*
- *Description → User/Help Text*
- *Column Component Priority (DDT) → “actual column order”*
- *Tablespace Name (DDT) → Tablespace Name on table implementation*

Method for discovering flows

- *Use test scenarios*
- *Look at the property values after the Transformers run*

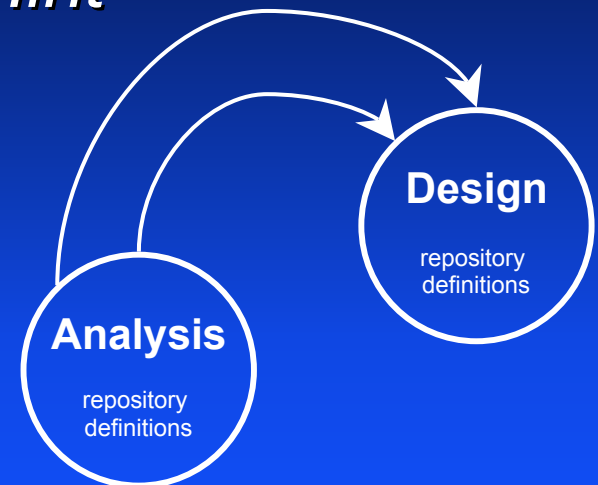


Analysis to Design

Attribute → Column

- *Name → Prompt & Source Attribute*
- *Format → Datatype & Display Type*
- *Description → User/Help Text & Description*
- *Comment → Comment & Hint*
- *Derivation → (nothing)*

Tip: Use the spreadsheet view of the property palette



Analysis to Design

Relationships → Keys

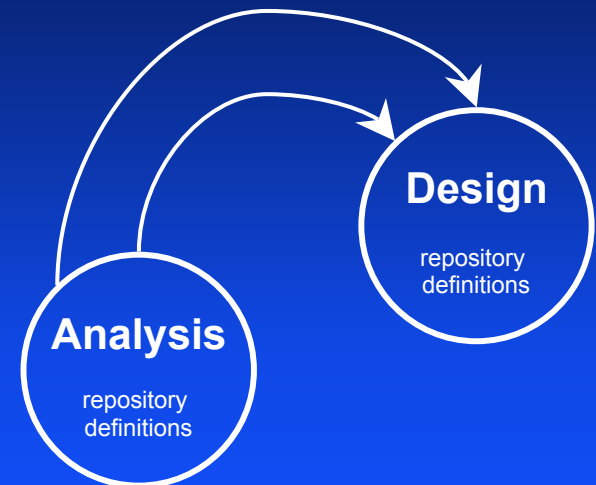
- *Copy to foreign key constraints and columns*

Subtype/supertype

- *Four implementation methods*
- *See paper for discussion*

Arc

- *Creates Server Model arc*
- *Triggers and Table API implement this*



Design to Build

Generate Database from Server Model

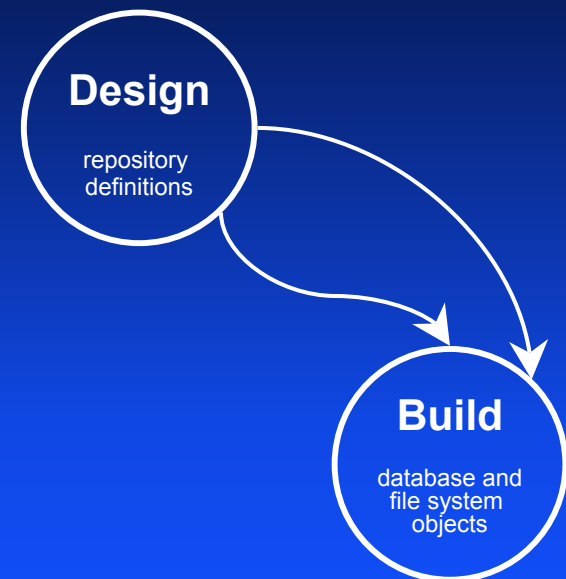
- *Output to CREATE scripts*
- *Server Model tab - no storage clause*
- *DB Admin tab - creates table implementation*
- *Generate Table API*

Client-code Generators

- *Output to Application*
- *Affected by many properties and object definitions*

Table, Column, Module, Table Usage, Bound Item, Preferences, Template

- *Generate Module Component API*



Build to Design - Database

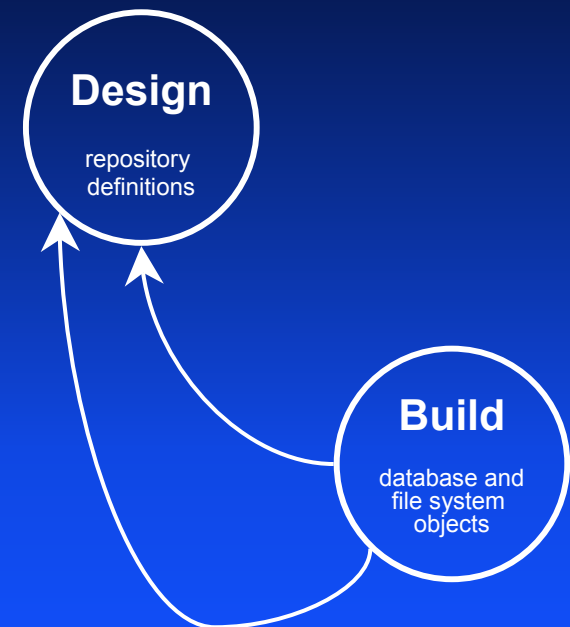
Capture Design of Server Model

Load definitions into repository from:

- *Online data dictionary - existing objects*
- *SQL script*
- *Other databases via ODBC*

Table definition

- *Table implementation*
- *Tablespace*
- *Storage Definition*
- *Database and Schema*



DEMO

Capture Design of Server Model

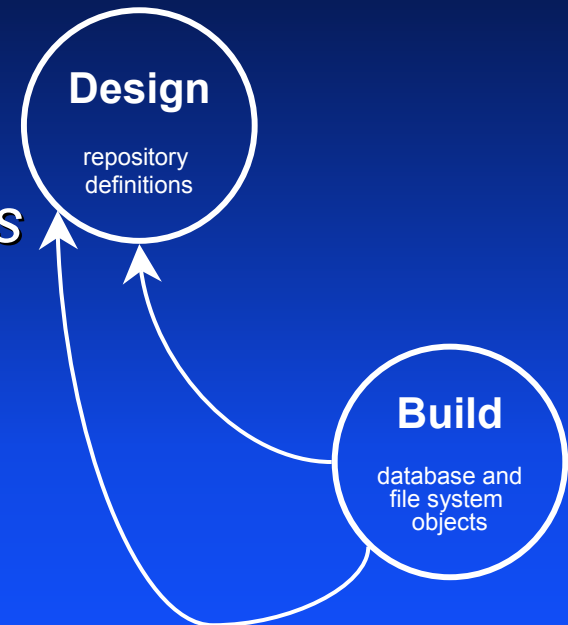
Build to Design - Application

Source code files → repository definitions

- *Module*
- *Module component*

Capture Design of Form

- *Table associations (links)*
 - *Master-Detail and Lookup tables*
- *Item prompts*
- *Item Type → Display Type*
- *All code in the proper triggers*



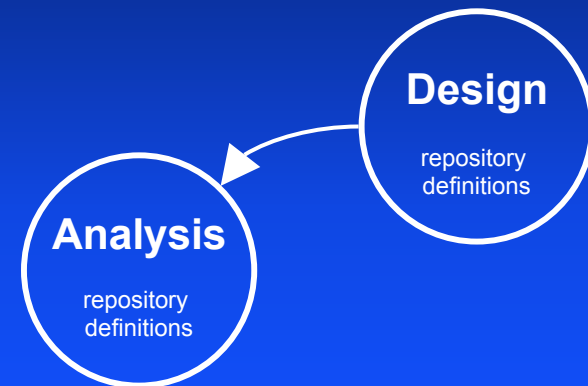
DEMO

Capture Design of Form

Design to Analysis

Table To Entity Retrofit

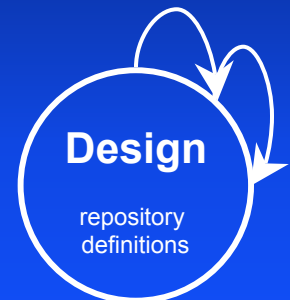
- *Table/Column/Constraint definitions → Entity/Attribute/Relationship definitions*
- *Cannot be used to reconcile differences between tables and entities*
- *Candidate tables are those without a Table Entity Usage*
- *Use it to create an ERD from legacy system*
 - *Uses Design Capture*



Design to Design

Generators change module definitions

- *Item width*
- *Missing rows displayed*
- *Dialog to ask if changing the definition is acceptable*
- *Autogenerated column set to non-enterable*
- *PK column set to non-updateable*





Modules - Navigator

Server Model | Modules | DB Admin | Distribution

- CTA[1]
 - Reusable Module Component
 - Modules
 - CTA MAIN MNU
 - CTAENROLL
 - CTAENTERINV
 - CTAGRADE
 - CTALIB
 - CTALIB2
 - CTALIBRARY
 - CTALOVZ
 - CTAPRINTINV
 - CTAREG
 - CTAREGPAY
 - CTAREGW
 - CTAREG_MNU
 - CTASETUPGRADE
 - CTASHOWZIP
 - CTASHZIP
 - CTAST
 - CTASTU
 - CTASTUDN
 - CTASTUREG
 - CTAUTIL

Design Capture is complete.

Design Capture is complete.

The changes that will be made to the Repository are now shown.
You can Save, Revert or Browse/Edit these changes.

Save Revert Browse/Edit Help



Forms Generator (Windows 95/NT) : Version 5.0.20.9.0 - Production on Tue Nov 03 22:20:23 1998
Copyright (c) 1995, 1998 Oracle Corporation. All rights reserved

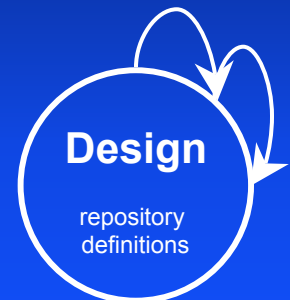
Capturing Module 'STUDENTS'
Capture of 'STUDENTS' Successful

Design and Application Logic Capture Complete

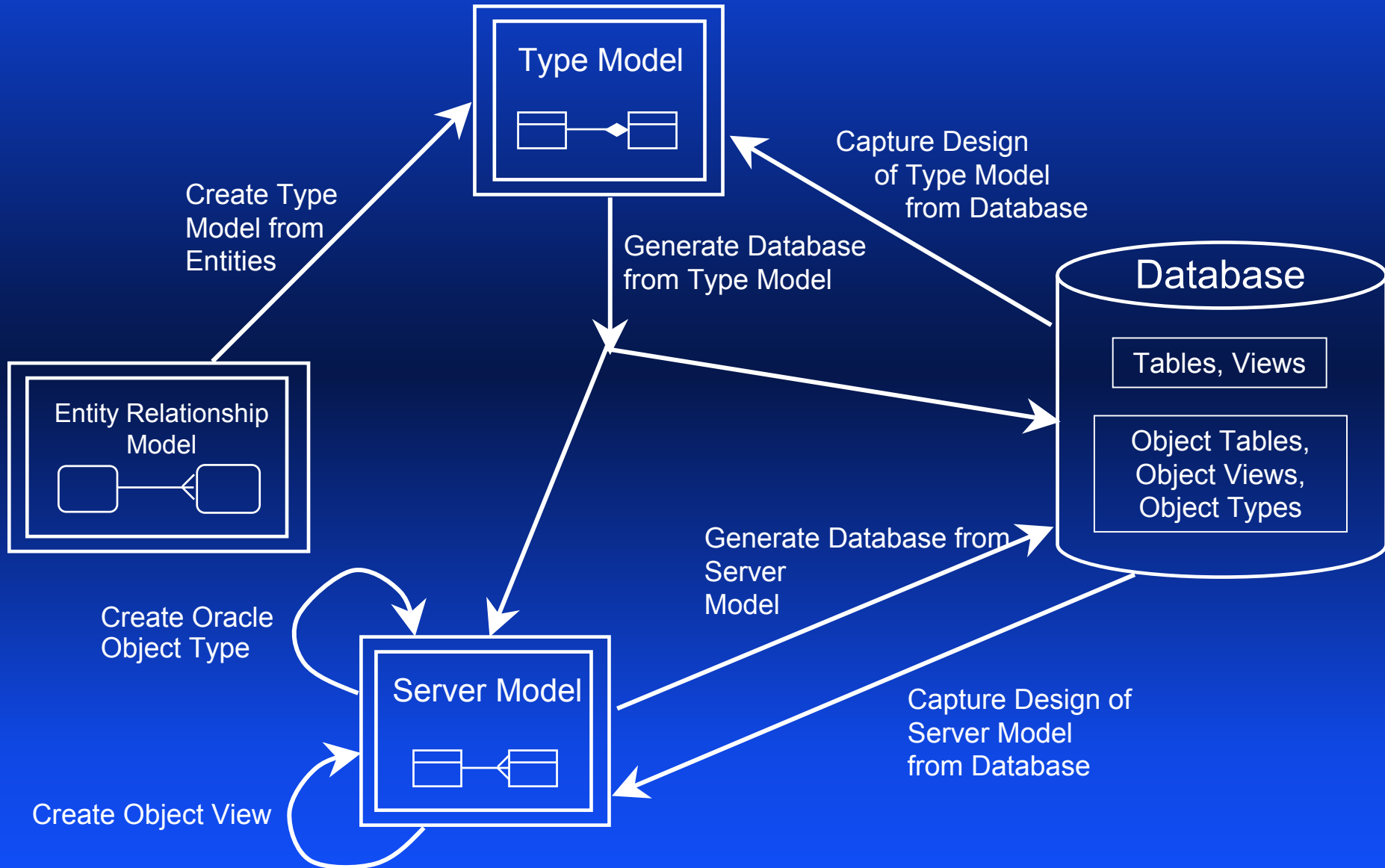
Design to Design

Object Database Designer (ODD)

- *Various transformations within the Design area*
- *Type Model to Server Model*
- *Type Model to Database*
- *Database to Type Model*
- *Entity Model to Type Model*
 - *Actually, Analysis to Design*



ODD Transformations



DEMO

Help System

Tip: New Information Flow Help in V.2

Help system has topics similar to tables in this paper

Perform an index search for headings

Example:

- Repository *[element]* properties used by Form Generator
- *[element]* is a definition like “module”

Repository module properties used by Form Generator

Related Topics

Module Property	Use to specify to Form Generator	Notes and Related Topics
Short Name	Determines the name of the form module, the .fmb file, and the .fmx file (if the Implementation Name property is set, this property is not used)	If this does not conform to Form Builder naming requirements, generation stops with an error. Included as graphic text in the generated form if CG\$MS is in the template.
Name	Included as graphic text in the generated form if CG\$MN is in the template	Module Name also used if no Top Title has been specified (i.e. to set the generated module's Title property, and when CG\$M1 is in the template).
Language	Verifies that the module has been specified for implementation as a Form Builder application.	Should be set to 'Developer/2000 FORMS'.
Purpose	Included as graphic text in the generated form if CG\$MP is in the template. Included in form level comments if the string 'PP' appears in the FRMCMT preference.	
Implementation Name	If set, overrides the Short Name property and determines the name of the form module, the .fmb file, and the .fmx file.	If this does not conform to Form Builder naming requirements, generation stops with an error. Included as graphic text in the generated form if CG\$M1 is in the template.

Help Topics: Designer/2000 Design and Generation



Contents | Index | Find

1 Type the word(s) you want to find

repository properties used by Form Generator

Clear

2 Select some matching words to narrow your search

... Generator
... Generator's
... Generators

Options...

Find Similar...

Find Now

Rebuild...

3 Click a topic, then click Display

Repository module properties set during design capture
Repository module properties used by Form Generator
Repository module properties used by Report Generator
Repository module properties used by WebServer Generator
Repository table usage properties set during design capture
Repository table usage properties used by Form Generator
Repository window properties used by Form Generator

66 Topics Found

All words, Begin, Auto, Pause

Display

Print...

Cancel

How to Use the Proceedings Paper

*Use the categories to identify the flows
you are interested in*

*Use the property mapping tables as
reference*

*Supplement this with the online help
system*

*Download the enhanced paper from our
web sites (see next slide)*

- Adaptation from “the book”*

Author Contacts

Fill out the eval - paper 323

Peter Koletzke

- *<http://www.mvsn.com>*
- *peter_koletzke@compuserve.com*

Oracle Designer Handbook, 2nd Ed

*Osborne McGraw-Hill, Oracle Press,
ISBN 0-07-882417-6*

- *co-authored by Dr. Paul Dorsey with lots of help from Douglas.*

Douglas Scherer

- *www.coreparadigm.com*
- *dscherer@coreparadigm.com*

