Achieving Great Web Performance Using ONLY SQL and PL/SQL

Dr. Paul Dorsey & Michael Rosenblum

Dulcian, Inc.

www.dulcian.com

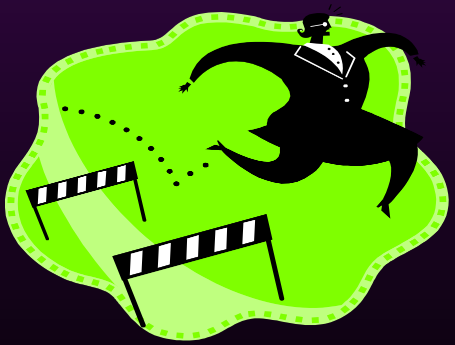
Sept 28, 2010

The Task

- Budget and Finance System for the government of Ethiopia
  - 1000 sites, 5000 users
  - 20 languages
  - Replace a legacy system
    - SQL Server => Oracle
    - Complex => simple architecture
    - No change in user functionality

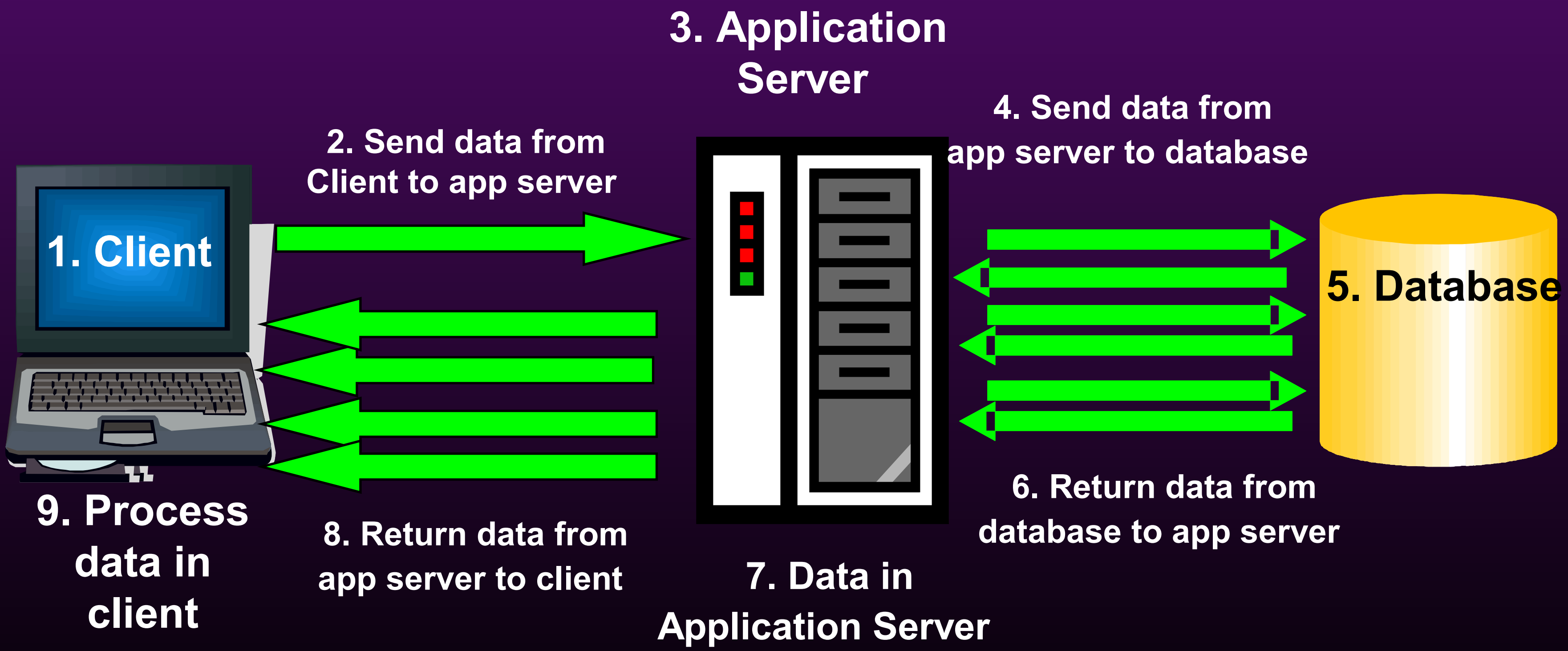
The Challenges

- Limited connectivity
- Large area (2 times the size of Texas)
- Limited IT skills of government employees
- No senior IT skills available in country
- Dirty data in source system
- Cultural differences

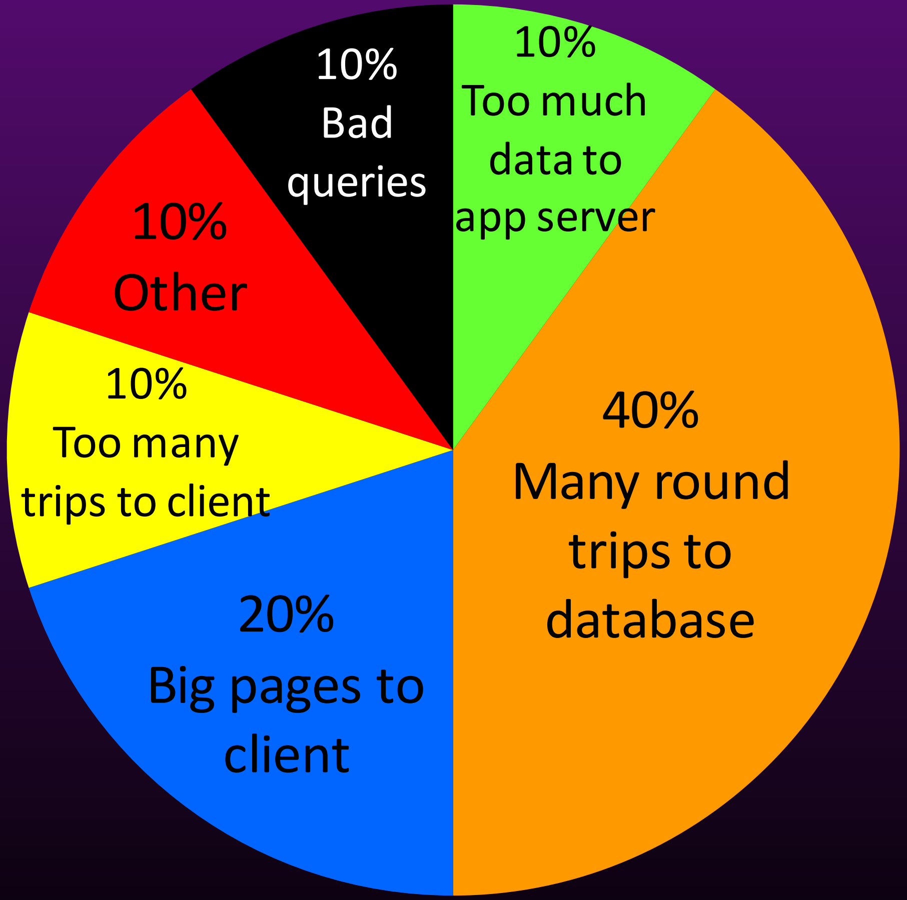


The Problem

- Everyone assumes infinite bandwidth.


3. Application Server
2. Send data from Client to app server
4. Send data from app server to database
1. Client
5. Database
9. Process data in client
8. Return data from app server to client
7. Data in Application Server
6. Return data from database to app server

Why is my web application slow?



Many round trips from application server to database

- Getters and Setters are problematic
  - Fannie Mae
    - 26.5 years to execute month-end routine
  - DOD
    - 60,000 round trips to populate 1 screen
  - USAF Reserve Recruiting
    - Batch routine
      - 20 minutes in Java
      - .2 seconds in PL/SQL

Send data from application server to database

Big Pages

- Web Center
  - "Some of our pages are less than 1MB."

"Mr. Page Bloat"

The Solution

1

- One round trip from database to application server per UI operation

2

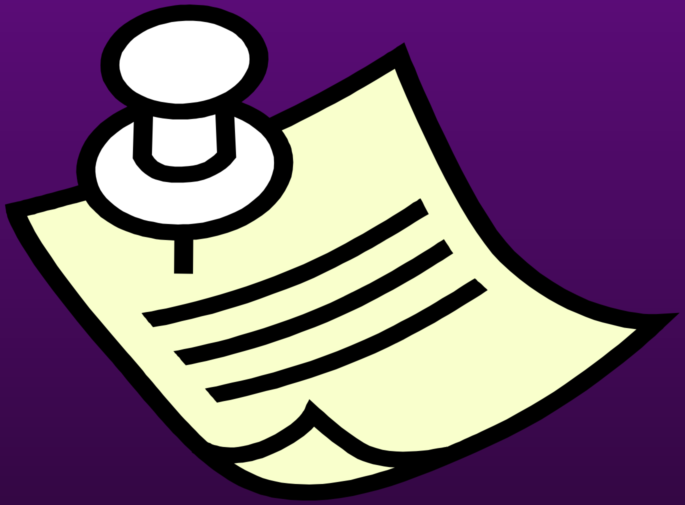
- Minimize page size

One round trip from application server to database (Implications)

- Thick database or no SQL
  - No context switch
- ALL user interface information in one place
  - Only way to reduce round trips to zero
- Ultra-thick database
  - Everything in the database

Minimize page size

- How small is small enough?
  - High bandwidth (>1MB/second)
    - 1 MB page is OK
  - Low bandwidth (5k/second)
    - 10K is the maximum
- Industry standard
  - Modern, cool, Web 2.0
    - >1MB
  - Basic HTML
    - 40K



Not Small Enough!!

What is possible?

- Logical description of page

```
<Page height = "200" ...>
  <Field height = "20" .../>
  <Field height = "20" .../>
  <Button label = "Save" .../>
</Page>
```

- UI Layout 4K
- Data 1K
- First time load = 5K
- Subsequent load = ≤1K

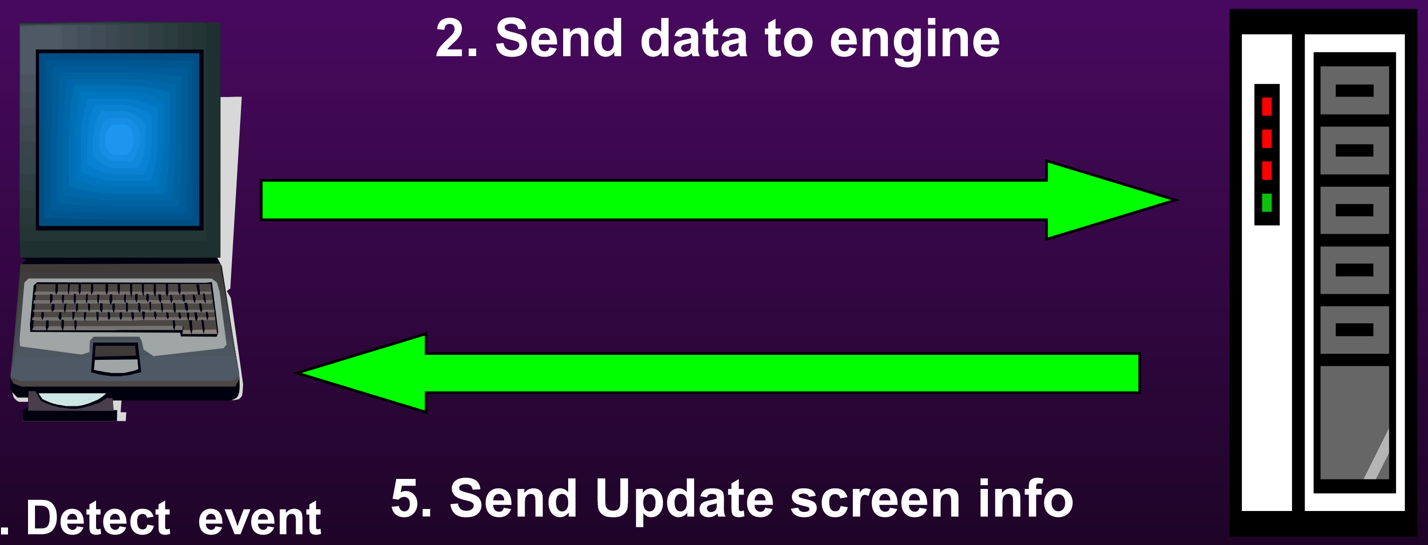
Implications for desired architecture

- 1) It doesn’t currently exist.
- 2) Forget industry standard.
- 3) Must keep complete copy of UI state in the database.
- 4) Super smart “browser” required
- 5) Application Server has minimal role.
- 6) Ultra-thick database
- 7) Minimal runtime logic sent to client

Other Constraints

- 1) Simple to learn/use
- 2) Productive
- 3) Functionally complete cool Web 2.0 pages
- 4) Rule-based
  - "The articulation of the rules is independent of the implementation of the rules."
- 5) UI tech stack-independent

The Solution:
Event/Action Framework (EAF)



Client

Engine

2. Send data to engine

5. Send Update screen info

1. Detect event

6. Update screen

3. Determine action
4. Process

What do we need?

- 1) Client
  - Event Detector
  - Action Interpreter
- 2) Server
  - Magic Engine
- 3) Interface Architecture
  - How to communicate between client and engine

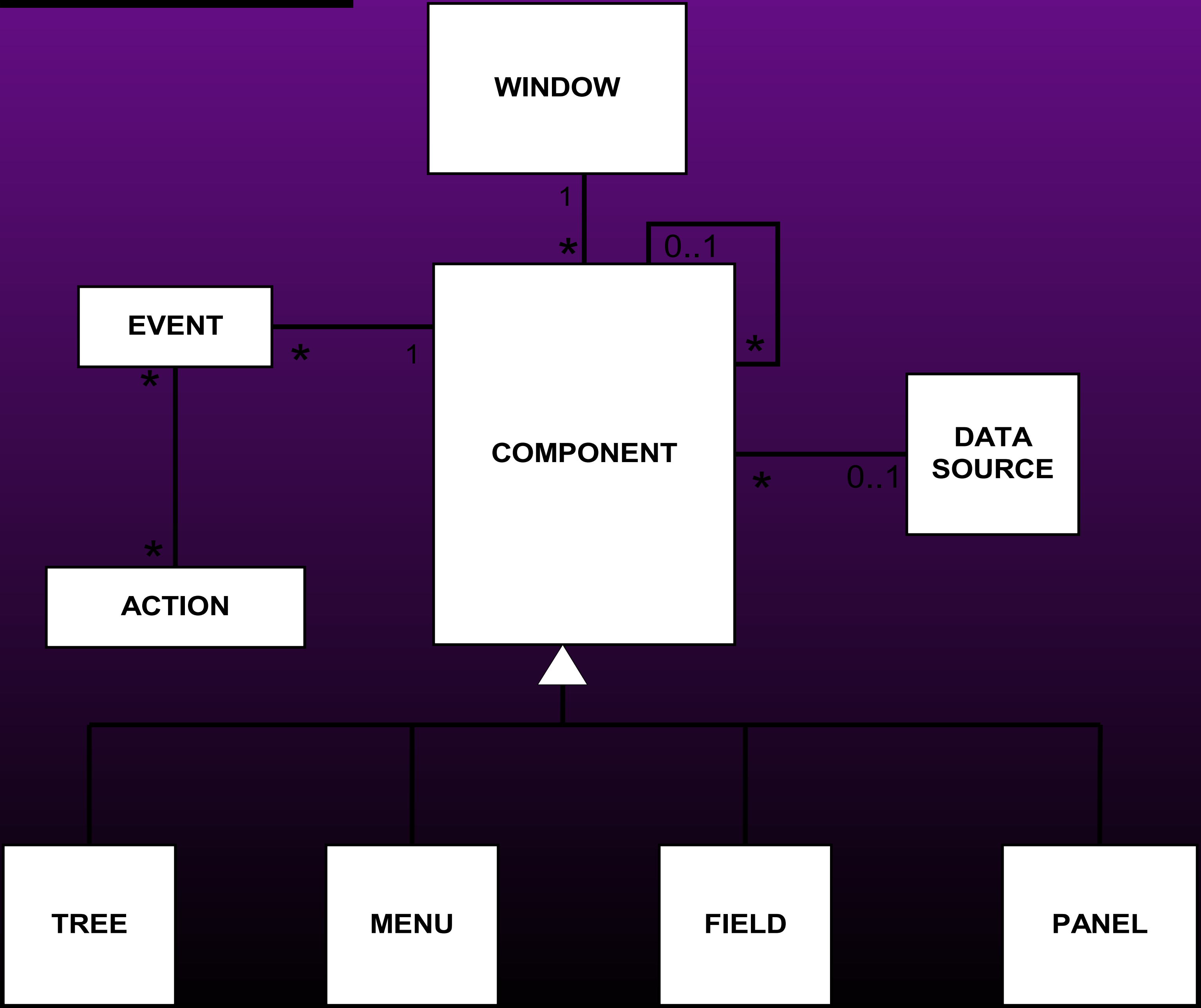
Interface Architecture

- XML for communication

```
<Screen>
   <Field . . ./>
   <Button . . ./>
<Screen>
```

Why are pages so BIG???

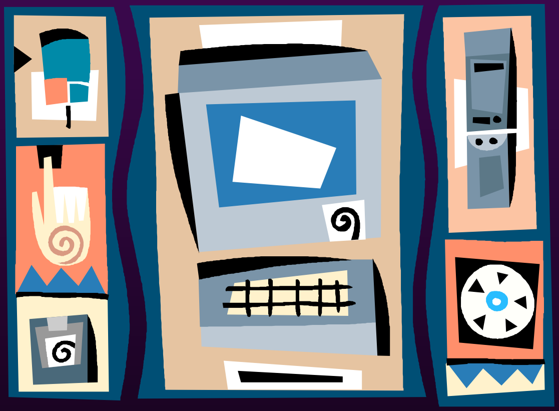
Repository Data Model



Client

- Big JavaScript library
  - ExtJS foundation for components
- Java

Do the formatting on the client!!



Server

- 1) Repository
- 2) Scripting Language
- 3) Runtime Engine
- 4) IDE

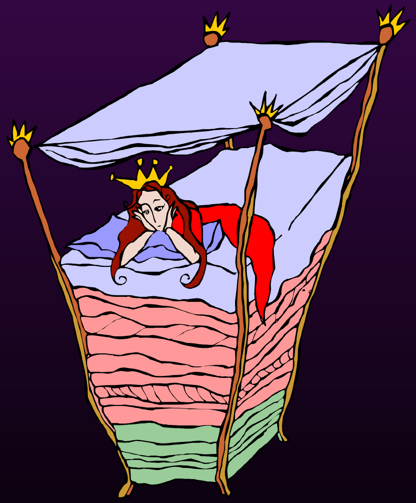
Advantages

- Easy to learn (easier than APEX)
- Client/Server quality on the web
  - 100% of Forms functionality implemented
- Rapid development (a little faster than Forms)
- Only SQL & PL/SQL required
- Fastest web applications ever
  - 10x -100x reduction in network traffic
- Deploy client/server or web (NO conversion cost)

But how???

Thick database techniques

- UI screens NEVER touch tables.
  - De-normalized views
  - Function-based views
- All complex data transformations in PL/SQL only!
- Effective utilization of:
  - BULK operations
  - CLOBs
  - XML types



De-Normalized Views

- The idea:
  - Convert relational data into something that will make user interface development easier.
  - Easiest way to separate data representation in the front-end from the real model.
- The solution:
  - Use a view with a set of INSTEAD-OF triggers.

De-Normalized view

```
create or replace view v_customer
as
select c.cust_id,
       c.name_tx,
       a.addr_id,
       a.street_tx,
       a.state_cd,
       a.postal_cd
from customer c
left outer join address a
   on c.cust_id = a.cust_id
```

INSTEAD-OF Insert

```
create or replace trigger v_customer_ii
instead of insert on v_customer
declare
  v_cust_id customer.cust_id%rowtype;
begin
  if :new.name_tx is not null then
   insert into customer
      (cust_id,name_tx,phone_tx)
    values
      (object_seq.nextval,:new.name_tx,:new.phone_tx)
   returning cust_id into v_cust_id;
  if :new.street_tx is not null then
   insert into address
      (addr_id,street_tx,state_cd, postal_cd, cust_id)
   values (object_seq.nextval,:new.street_tx,
    :new.state_cd,:new.postal_cd, v_cust_id);
  end if;
end;
```

Function-Based Views (1)

- Case:
  - Complex search engine
    - About 20 different filtering criteria
    - Applicable to different tables
    - Large data volume

- Problem:
  - Unpredictable performance results in a single SQL query.
- Solution
  - Function-based view with dynamic SQL under the hood.

Function-Based Views (2)

- A. Create an output object with corresponding collection.

```
CREATE type search_ot as object
(Name_TX Varchar2(50),Phone_TX varchar2(20)…)
CREATE type search_nt as table of search_ot;
```

- B. Create a function to return collection all search criteria become input variables

```
CREATE OR REPLACE FUNCTION f_search_tt
  (i_name_tx varchar2, i_phone_tx varchar2, …)
RETURN search_nt
IS
  v_tt search_nt:= search_nt();
BEGIN
  RETURN v_tt;
END;
```

Function-Based Views (3)

- Use Dynamic SQL build the query

```
FUNCTION f_search_tt IS
  v_sql_tx varchar2(32000);
BEGIN
  v_sql_tx:='select search_ot(...) '||chr(10)
            'from ... '||chr(10)
            'where ...';

  if i_name_tx is not null then
    v_sql_tx:=v_sql_tx||
      ' and cust.name_tx like ''%'||i_name_tx||'%'' '
  end if;
  ...

  execute immediate v_sql_tx bulk collect into v_tt;
  ...
END;
```

Function-Based Views (4)

- Give code to developers

```
select name_tx, address_tx, phone_tx, …
from table(
           cast(f_search_nt
                  (:1, -- name
                   :2, -- phone
                    …
                  )
           as search_nt)
           )
```

Conclusions

- We CAN do better
- We do not need…
  - Complex architectures
  - FAT pages
  - Lots of big servers
- The keys…
  - Rules approach
  - Ultra thick database
  - All UI logic and processing in the server

Share your Knowledge:
Call for Articles/Presentations

- Submit articles, questions, … to

IOUG – The SELECT Journal

select@ioug.org

Reviewers wanted

ODTUG – Technical Journal

pubs@odtug.com

Dulcian's BRIM® Environment

- Full business rules-based development environment
- For Demo
  - Write "BRIM" on business card

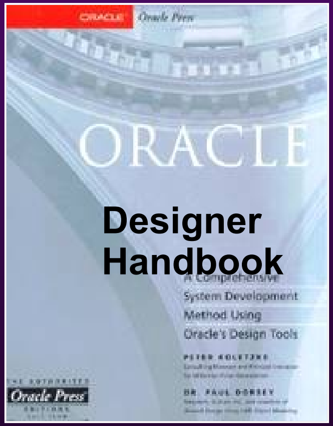
Contact Information

- Dr. Paul Dorsey – paul_dorsey@dulcian.com
- Michael Rosenblum – mrosenblum@dulcian.com
- Dulcian website - www.dulcian.com

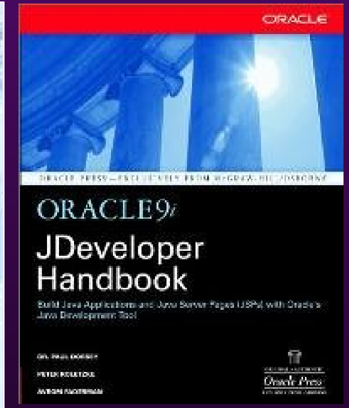
Design Using UML Object Modeling

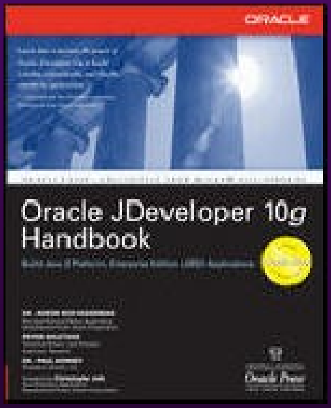
ORACLE
Oracle Press
Development Standards and Techniques
PETER KOLETZKE
DR. PAUL DORSEY
COVERS DEVELOPER RELEASES 2.0 THROUGH 6.0

Developer Advanced Forms & Reports


ORACLE
Oracle Press
A Comprehensive System Development Method Using Oracle's Design Tools
PETER KOLETZKE
DR. PAUL DORSEY

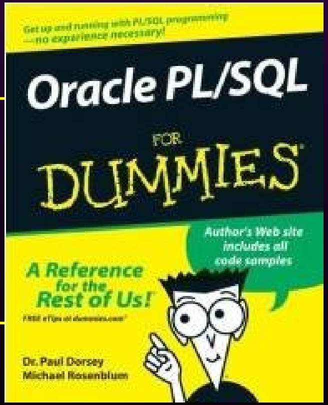
Designer Handbook





Latest book:

Oracle PL/SQL for Dummies


Get up and running with PL/SQL programming —no experience necessary!
Oracle PL/SQL
FOR
DUMMIES
Author's Web site includes all code samples
A Reference for the Rest of Us!
Dr. Paul Dorsey
Michael Rosenblum