Analysis and Modeling of Evolving Database-centric Web Applications

Sista Venkata Madhava Krishna[+]
IIT Guwahati

Karnati Satyadeep[+]
IIT Guwahati

Abhishek Biswas
Old Dominion University

Jagannathan Srinivasan
Oracle Inc.

[+]Work done as part of a summer internship at
Sarada Research Labs, Bangalore

Prepared For
16th International Conference on Management of Data
COMAD 2010, Nagpur, India
Date: December 9, 2010

Outline

- Database Centric Web Applications
  - *Challenges in Application Evolution Tracking*
  - *Ideal Web Application "Diff" Utility*
- Evolution Management Model
  - *Evolution Hierarchies*
  - *Modeling Application Evolution*
- Evolving Application Synopsis Tool
  - *The Oracle APEX Environment*
  - *Tour of EAST*
- Experimental Study
  - *Application Evolution Characteristics*
  - *Experiment Results*
- Conclusion & Future Work

Database Centric Web Applications

- Significant Attributes
  - *Rapid Application Development (RAD)*
  - *Structured Hierarchy*
    - Application – Web Pages – Page Regions – Region Items
  - *Agile Software Development*
- Challenges in Evolution Tracking
  - *RAD software development*
  - *No component level versioning in the development tools*
  - *Database object dependencies not managed by the development environment.*

Definitions

- Page
- Page Region
- Region Items
- Process (Event handlers)

Key Contributions

- Propose an architecture to automatically analyse and model synopsis of evolving database-centric web applications
- Web application region similarity measures for equivalence determination and change tracking
- The EAST tool and its use in studying evolving applications that demonstrates the usefulness of our approach

Related Work

- Popular source code diff
  - *Based on file comparison algorithms*[4]
  - *Syntactic & Semantic comparison of programs*[5,6]
- Change in document structure[8]
  - *XML structure change*[9,11]
- Database schema versioning
  - *Track evolution in database schema*[7]
  - *Challenging to track database dependencies*

Outline

- Database Centric Web Applications
  - *Challenges in Application Evolution Tracking*
  - *Ideal Web Application "Diff" Utility*
- Evolution Management Model
  - *Evolution Hierarchies*
  - *Modeling Application Evolution*
- Evolving Application Synopsis Tool
  - *The Oracle APEX Environment*
  - *Tour of EAST*
- Experimental Study
  - *Application Evolution Characteristics*
  - *Experiment Results*
- Conclusion & Future Work

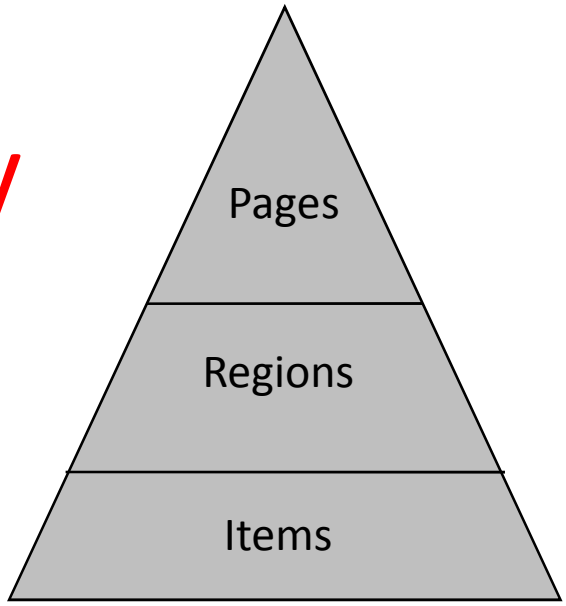
Evolution Management Model

- Evolution tracking model based on MVC Architecture
- Two basic evolution hierarchies
  - *View Hierarchy: Model evolution in user interface*
  - *Model Hierarchy: Model evolution in business logic*
  - *Hierarchies are built bottom up based on the contains relationship.*
- Schema Dependency Evolution Hierarchy Analysis
  - *Model dependency of application components on database schema objects*

Two Step Change Tracking

- *Establish Equivalence*
  - *Identify matching components from the two versions of the application*
  - *Based on matching of static attributes or similarity score based matching*
  - *Computation is done top down*
- *Change status tagging*
  - *Each component is tagged modified inserted or deleted*
  - *Tags of container component depend on tags on contained component*
  - *Therefore, Computed bottom up*

View Hierarchy



- Tracks changes in user interface
  - *Page Level Matching*
    - *Matching is based upon unique identifier*
    - *Can be implemented by score based matching*
  - *Region Level Matching*
    - *Compared based on pair wise region similarity score*
    - *m:n matching is allowed based on score threshold*
  - *Item Level Matching*
    - *Items are matched by item names*
    - *Item names are unique in a page as they are references and de-referenced*

Pair-wise Region Similarity Score

- Region similarity score computation
  - *Weighted average of four scores*
    - *1. Region Type Score*
      - » *Boolean similarity of region type*
    - *2. Region Name Score*
      - » *Threshold based edit distance*

$$\phi_{RgName} = \max\left(1 - \frac{e(r_1.name, r_2.name)}{T_n + 1}, 0\right)$$

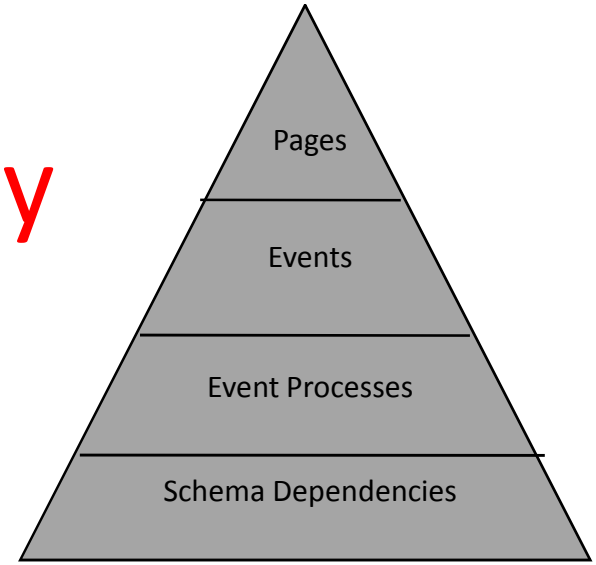
    - *3. Region Items Count Score*
      - » *Similarity based on common/added/deleted items*
      - » *a-items in region$_1$; b- items in region$_2$*
      - » *c-common items*

$$\phi_{RgItemCount} = \begin{cases} 0 \ \ if \ \ a=b=c=0 \\ \dfrac{2c}{a+b} \quad otherwise \end{cases}$$

    - *4. Region Source Score*
      - » *Change in source code*

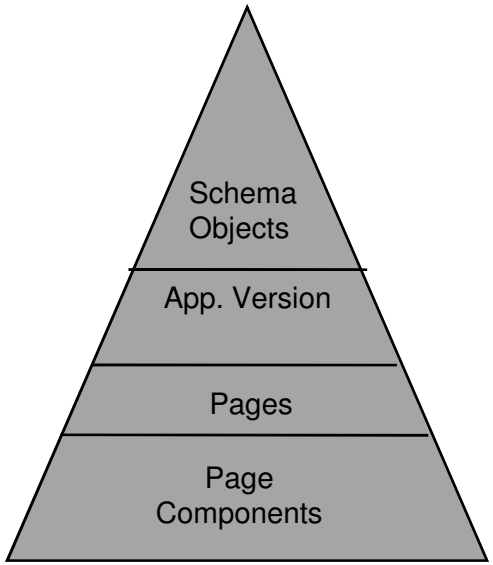
$$\phi_{RgSrcDiff} = \begin{cases} 0 & if \quad \#text_1 + \#text_2 = 0 \\ 1 - \dfrac{\#deleted + \#added}{\#text_1 + \#text_2} & otherwise \end{cases}$$

Model Hierarchy



- Used to analyze change in back end code.
  - *Events are pre-defined by application environment*
  - *Event handler processes are compared*
    - *Matching is performed based on name*
    - *Source code diff is performed for change tagging*

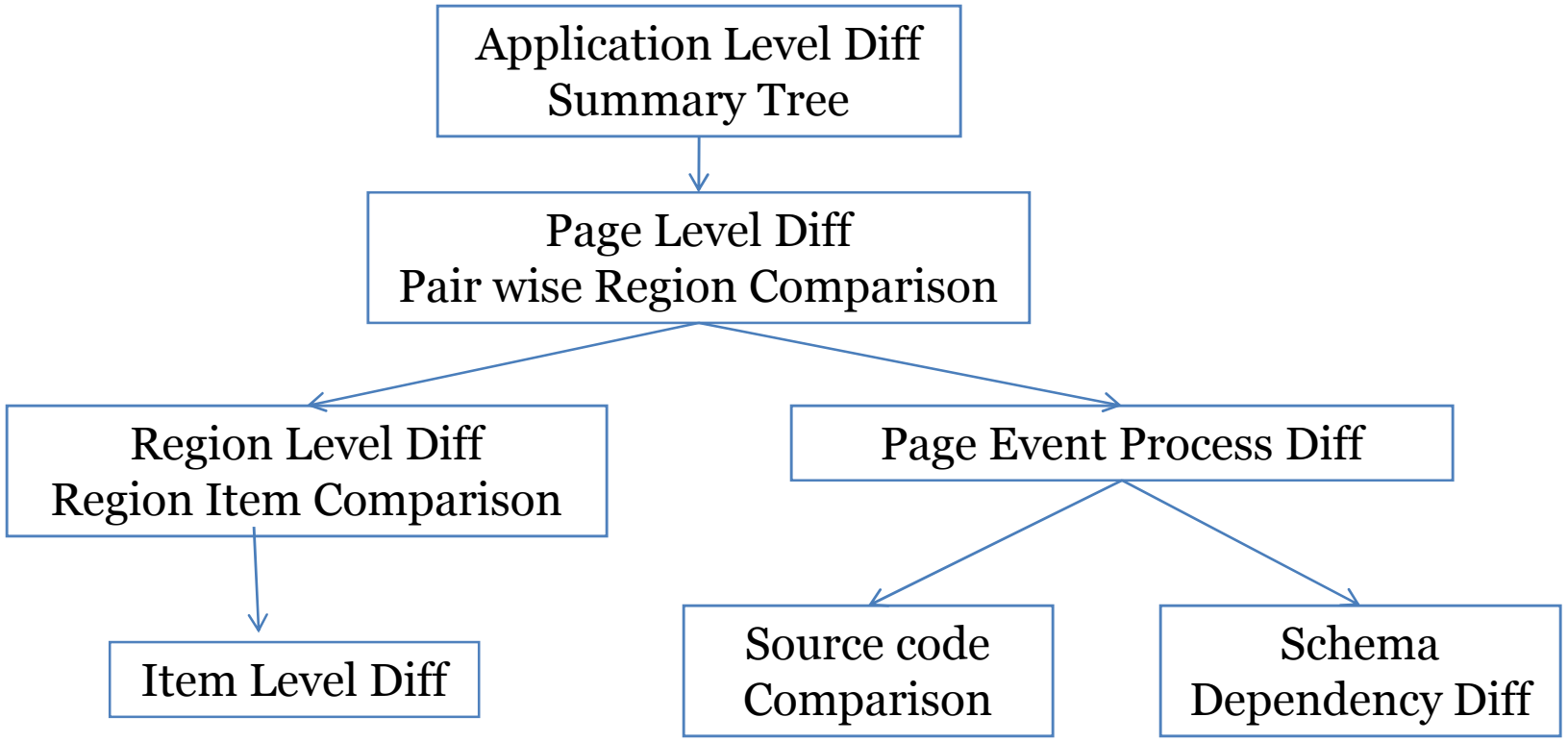
Schema Dependency Evolution Hierarchy Analysis



- Tracks change in dependency of application components on schema objects
  - *A list of schema references is compiled for each schema object*
  - *The lists from the two versions are compared*

EAST-Evolution Visualization Model

- Hierarchical drill down visualization of user interface



Outline

- Database Centric Web Applications
  - *Challenges in Application Evolution Tracking*
  - *Ideal Web Application "Diff" Utility*
- Evolution Management Model
  - *Evolution Hierarchies*
  - *Modeling Application Evolution*
- Evolving Application Synopsis Tool
  - *The Oracle APEX Environment*
  - *Tour of EAST*
- Experimental Study
  - *Application Evolution Characteristics*
  - *Experiment Results*
- Conclusion & Future Work

Oracle APEX Environment

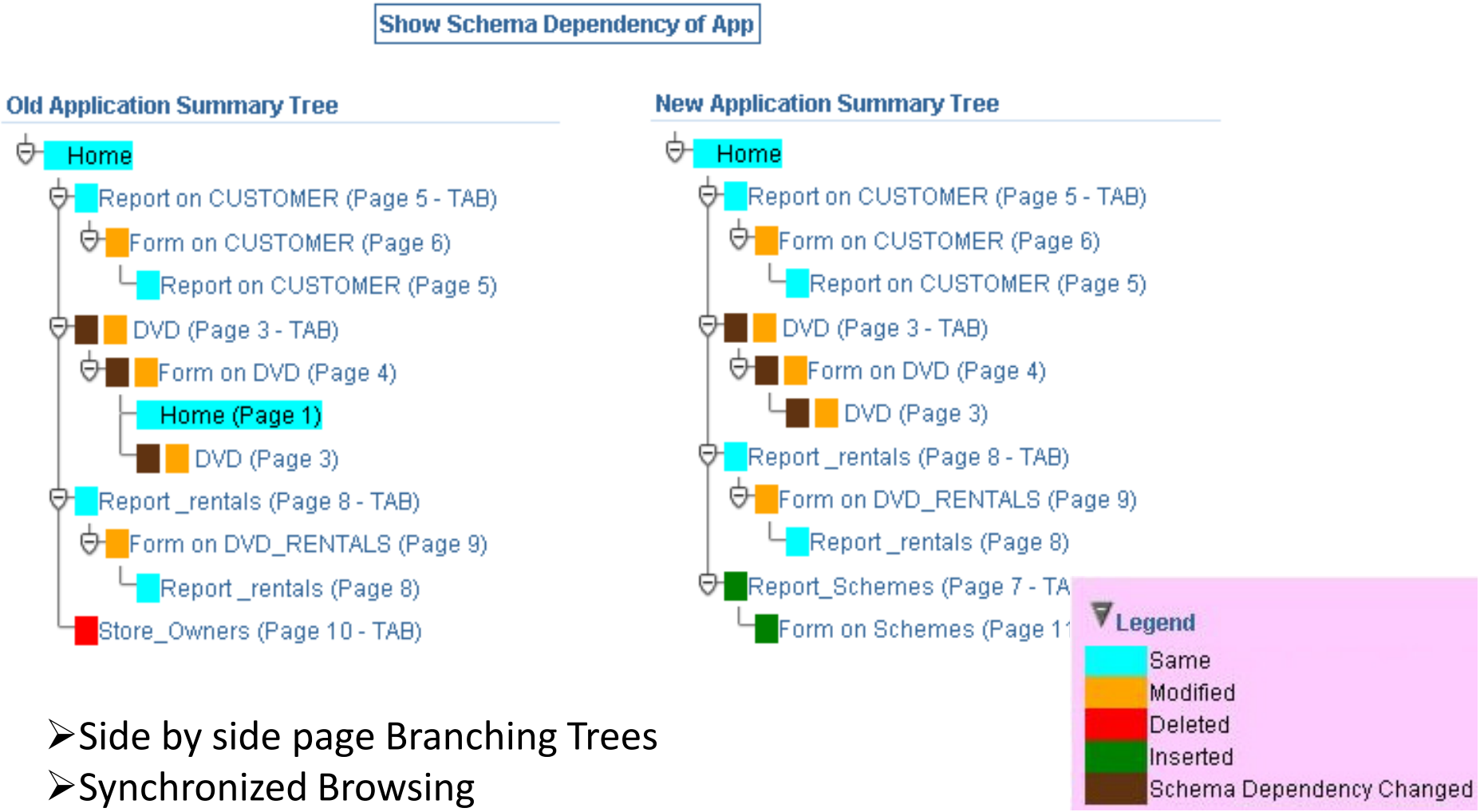
- Oracle Application Express (APEX) is a Rapid Application Development tool, that allows building a Database Web Application declaratively using a wizard.

- APEX stores all the *Metadata* for an application in Oracle Database, and makes this metadata available to users through an exhaustive set of *Views*.

- The SQL Queries & Process Logic specified using PL/SQL.

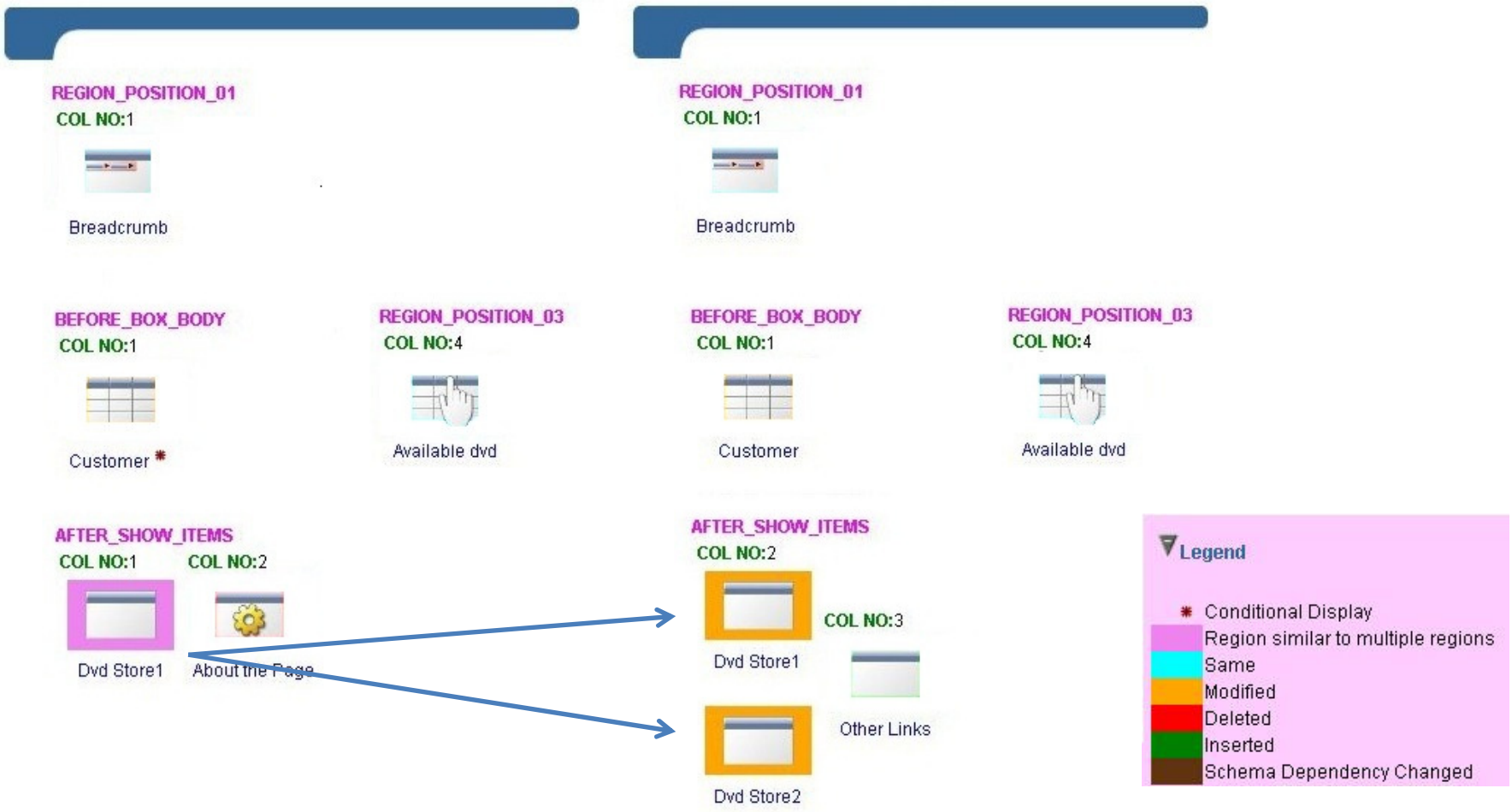
- The **EAST** has been built to leverage the APEX Views and APEX User interface components.

1. Application Summary Trees



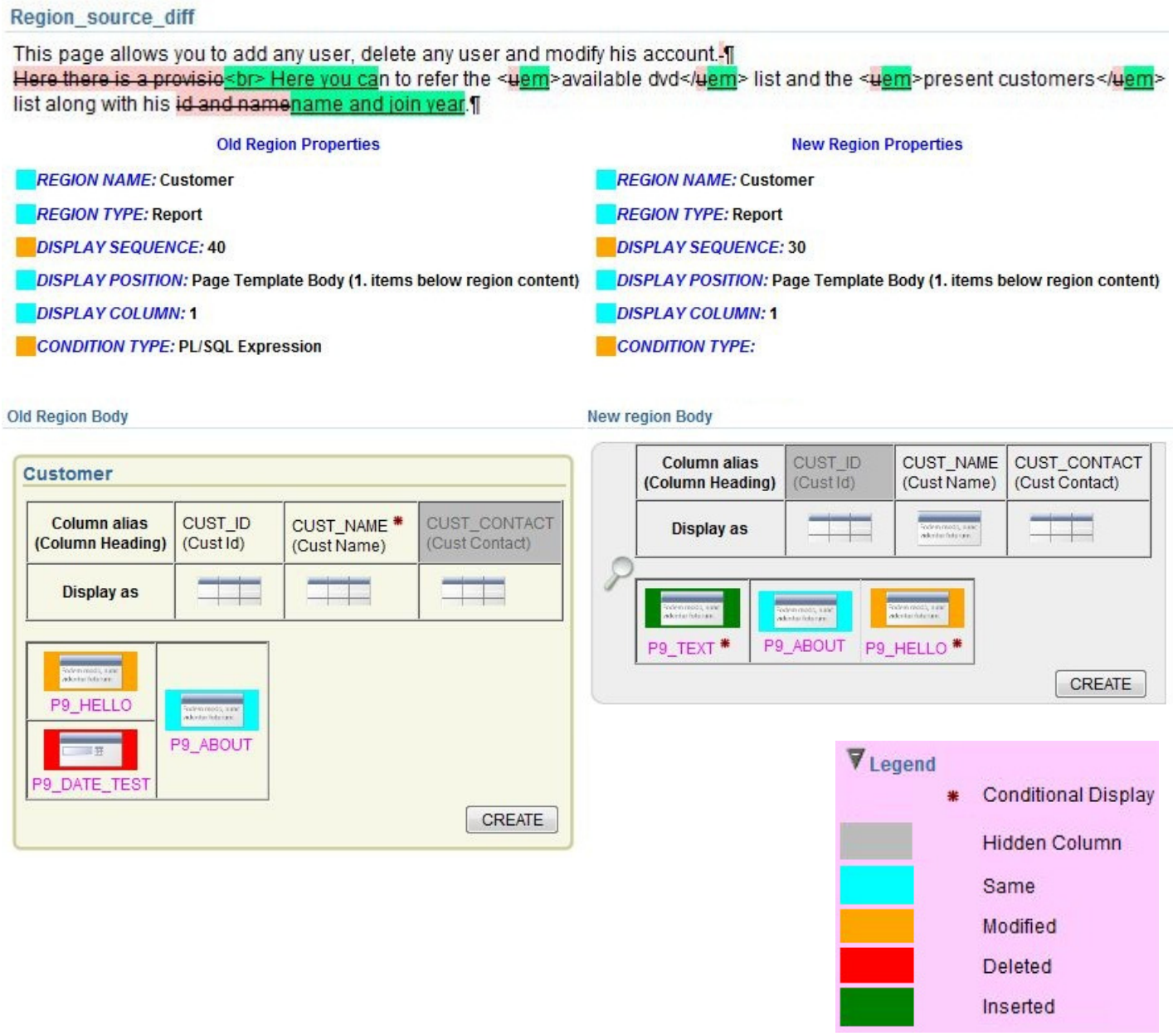
- Side by side page Branching Trees
- Synchronized Browsing
  - *The differences in branching transitions are easy to examine*
- Schema Dependency changes highlighted

2. Page Level Diff



- Page theme and template with regions in respective positions, showing the snapshot of the page.
- Multiple region similarity
  - *Caused due to clone and edit operations by developers*

3. Region Level Diff



- Source diff
  - Character based – suited for source code
- Properties diff
- Content diff
- Layout diff
  - Position and template

4. Process Diff

Page Process Diff

PROCESS_NAME	PROCESS_TYPE ▲	STATUS
reset_edit_print_opd1	Clear Cache for Items (ITEM,ITEM,ITEM)	
reset_edit_print_opd	Clear Cache for Items (ITEM,ITEM,ITEM)	
reset_new	Clear Cache for Items (ITEM,ITEM,ITEM)	
use details	PL/SQL anonymous block	
test OLD	PL/SQL anonymous block	
show	PL/SQL anonymous block	

Legend

	Modified
	Inserted
	Deleted

- Single tabular format
- Drilldown to diff of each process.

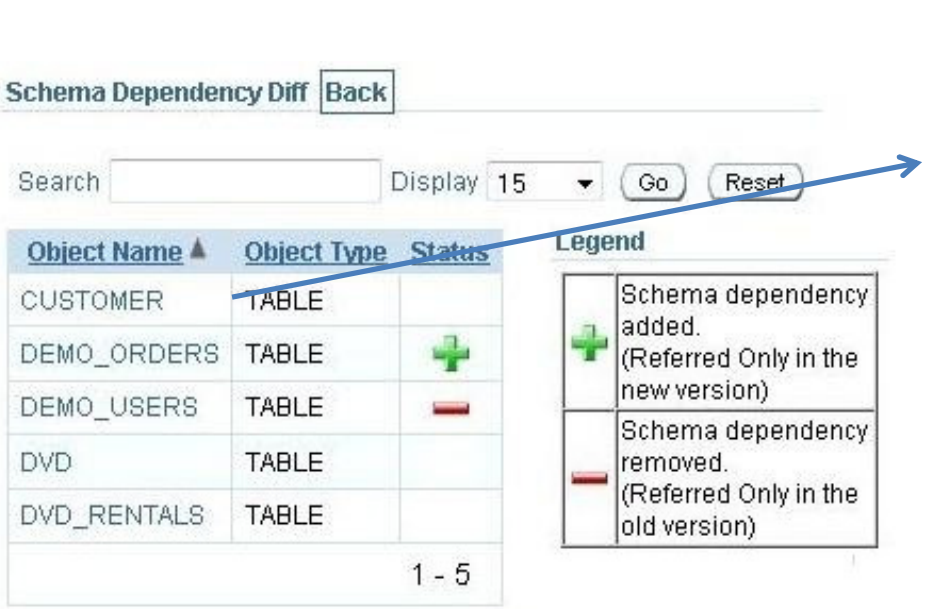
Process Properties : reset_new

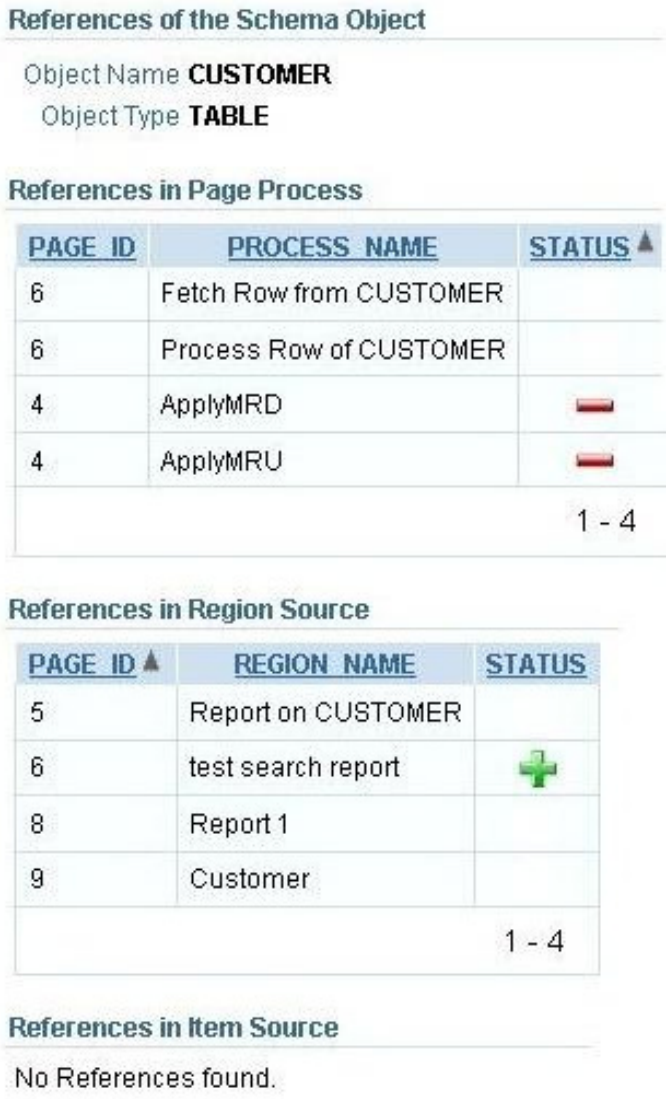
Property	Old Process	New Process
EXECUTION_SEQUENCE	55	55
PROCESS_POINT	On Submit - After Computations and Validations	On Submit - After Computations and Validations
WHEN_BUTTON_PRESSED	Reset_New	Reset_New
CONDITION_TYPE		
RUN_PROCESS	Once Per Page Visit (default)	Once Per Page Visit (default)

Process Source Diff

P2_NAME,P2_PATIENT_ID,P2_GENDER,P2_AGE,P2_MONTHS,P2_DAYS_AGE,P2_ROOM_NO,P2_FREE ¶

5. Schema Dependency Diff

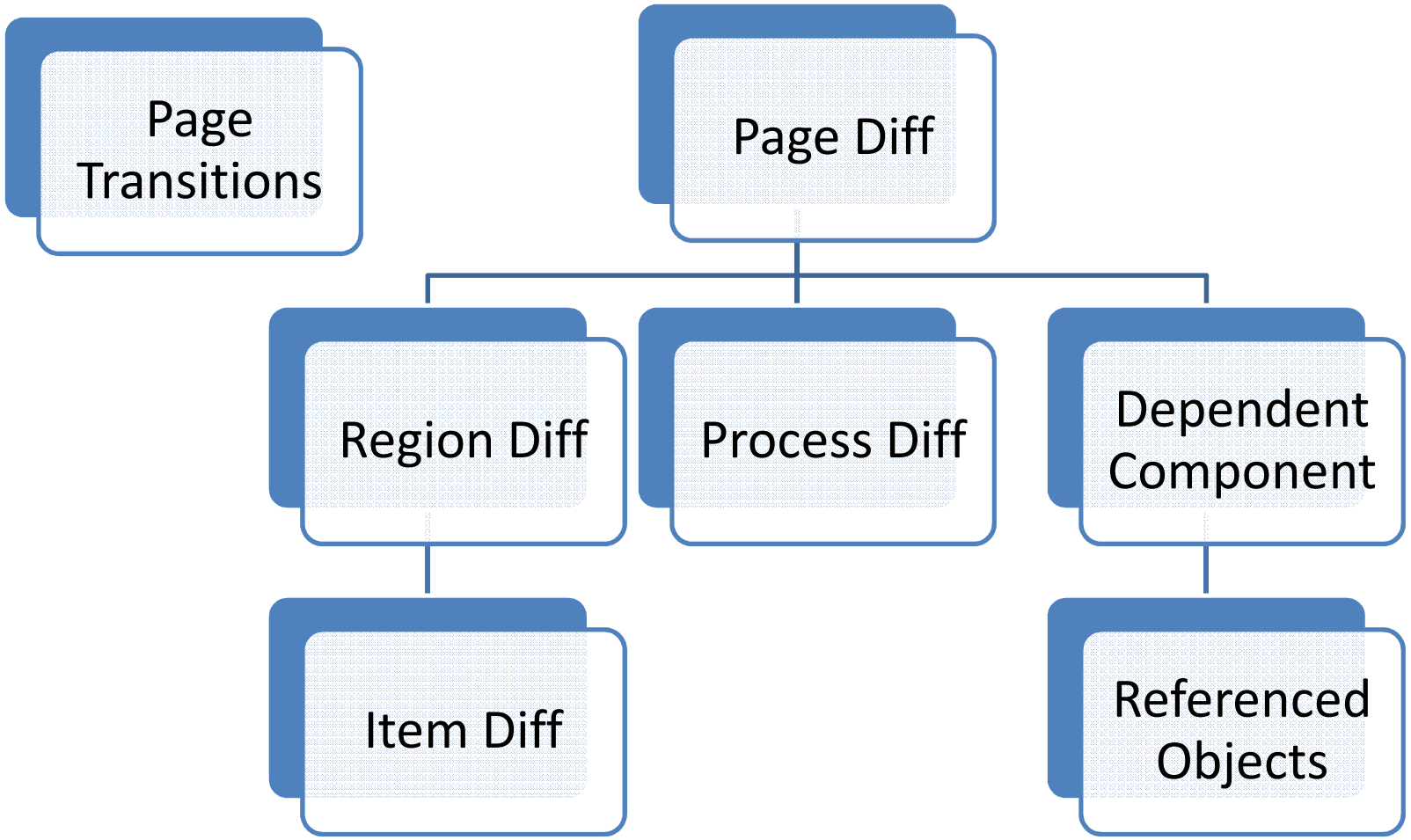




- Single tabular format

- Drilldown to show the list of all application components referencing a schema object

Database schema design



Database schema design

- Summary tree:
  - *Page Transitions table:* Page id's along with the parent page ids.
- Component Diff : (Pages, Regions, Items, Processes)
  - *Component Diff tables*: Equivalent component Id's and their modification status.

APP1_ID	APP2_ID	PAGE1_ID	PAGE2_ID	STATUS
110	111	10	-1	deleted
110	111	-1	7	inserted
110	111	-1	11	inserted
110	111	1	1	same
110	111	3	3	modified
110	111	4	4	modified
110	111	5	5	same

- Schema Dependency Diff :
  - *Dependent Component table*: Component id's + unique dependent id's
  - *Referenced Object table* : Ids of the referenced schema objects + dependent id's. (FKey)

Outline

- Database Centric Web Applications
  - *Challenges in Application Evolution Tracking*
  - *Ideal Web Application "Diff" Utility*
- Evolution Management Model
  - *Evolution Hierarchies*
  - *Modeling Application Evolution*
- Evolving Application Synopsis Tool
  - *The Oracle APEX Environment*
  - *Tour of EAST*
- Experimental Study
  - *Application Evolution Characteristics*
  - *Experiment Results*
- Conclusion & Future Work

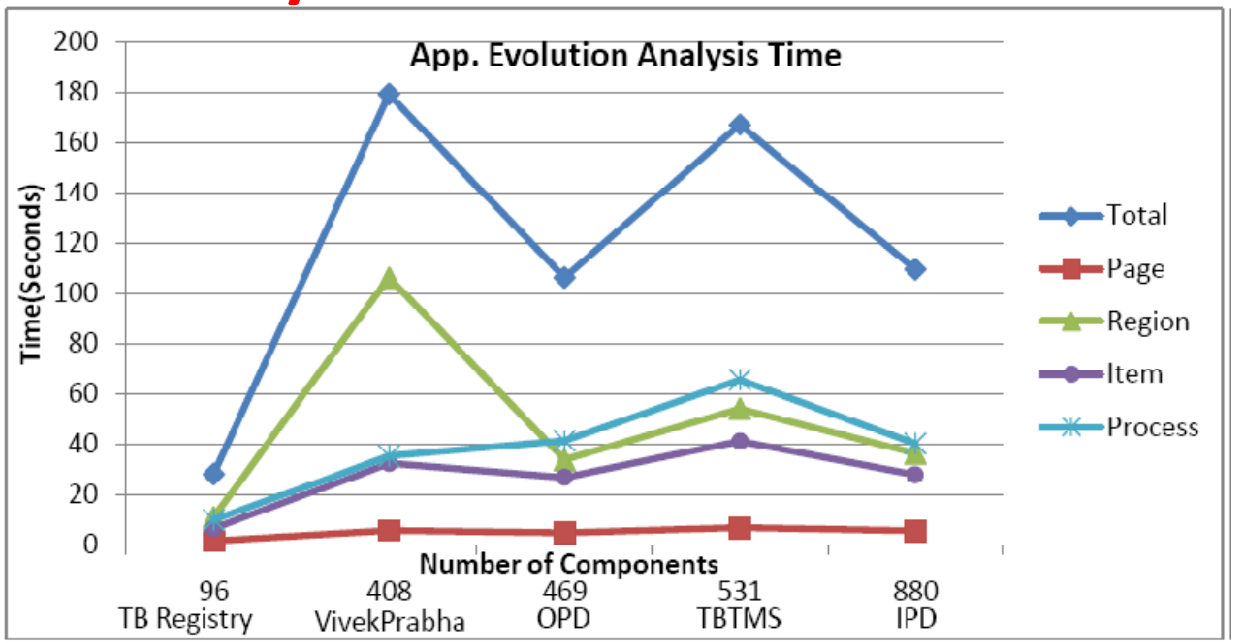
Applications Analyzed

	Version Diff (in months)	*App. Maturity (Old,New)	Pages (Old,New)	Regions (Old,New)	Items (Old,New)
TB Registry	0.75	(60,85)	9,9	32,36	51,51
OPD	12.50	(90,95)	27,37	106,136	228,296
IPD	12.50	(82,89)	35,57	93,229	225,594
Vivek Prabha	7.50	(85,100)	41,41	117,117	250,250
TBTMS	1.50	(65,75)	44,44	158,163	317,324

***Application Maturity** in % (as rated by the developer of application)

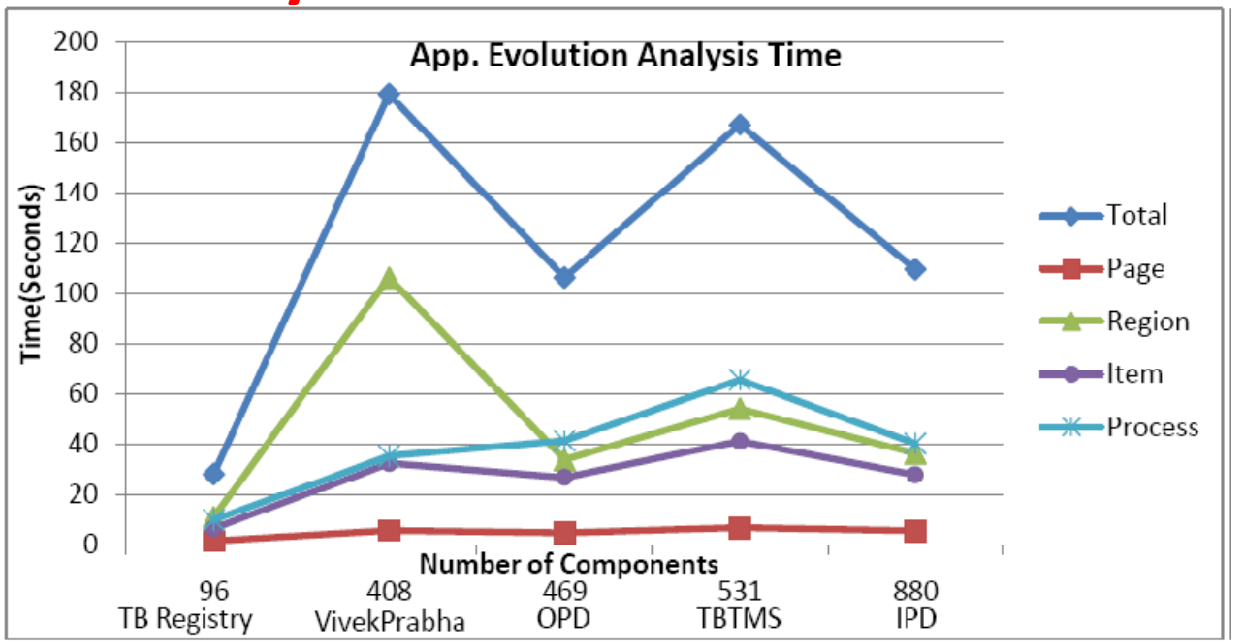
- Analyzed applications developed by Sarada Research Labs, all of which are already deployed.
- Wide span of evolution range of applications
- Negligible evolution (Vivek Prabha) to significant evolution (IPD)

Analysis Time Overheads



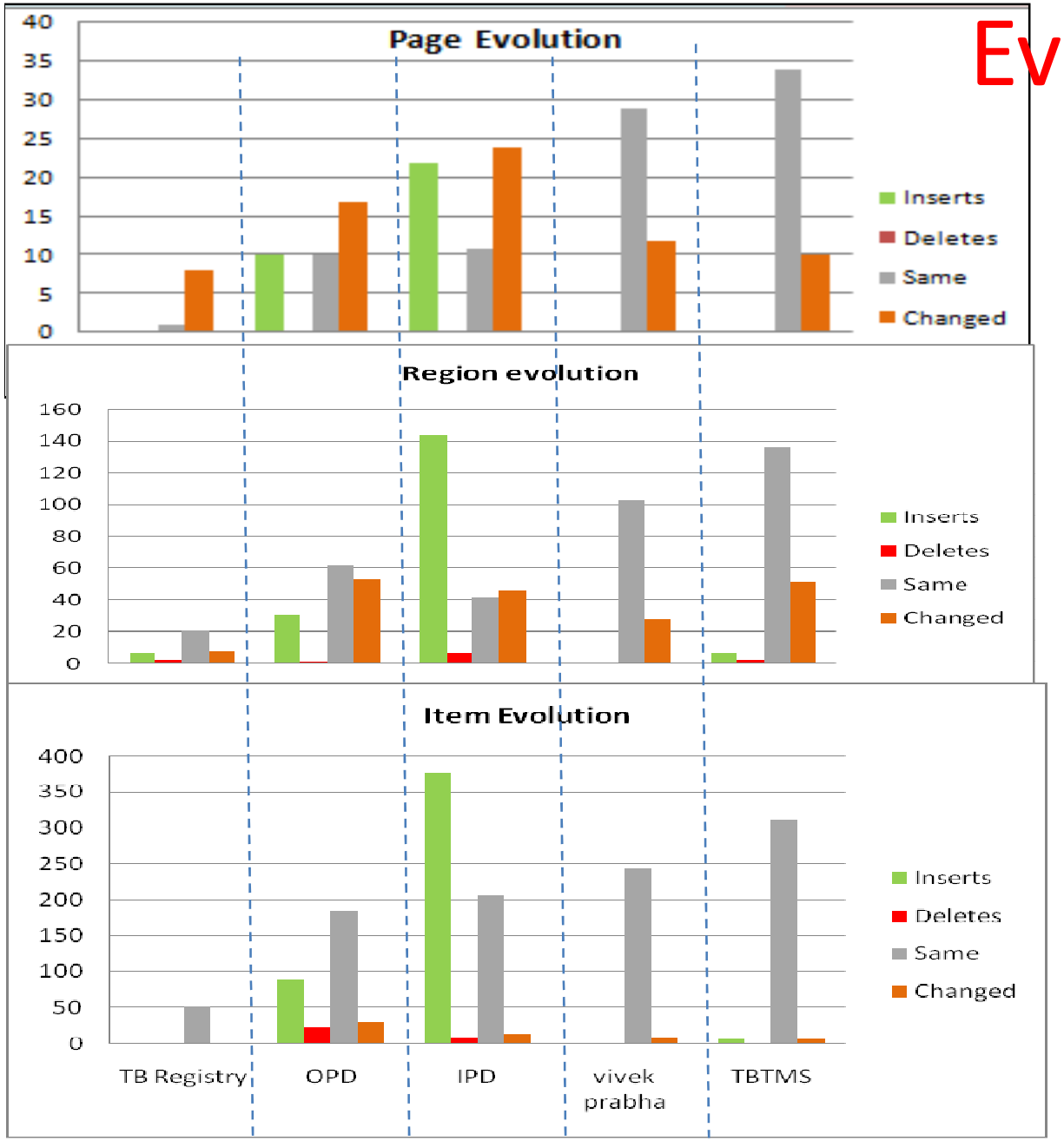
- Major overhead is establishing equivalence.
- Comparing two equivalent components is faster(once equivalence is established).
- Page, Item equivalence
  - established by their unique ids or names - takes less time.
- Region equivalence
  - need to compare properties for each region pair - takes a lot of time.

Analysis Time Overheads



- Regions: The presence of exactly similar regions raises analysis time.
  - ➢ involves comparing all the properties.
- Process : Analysis time dominates overhead cost
  - ➢ Costlier source code diff.

Evolution Trends



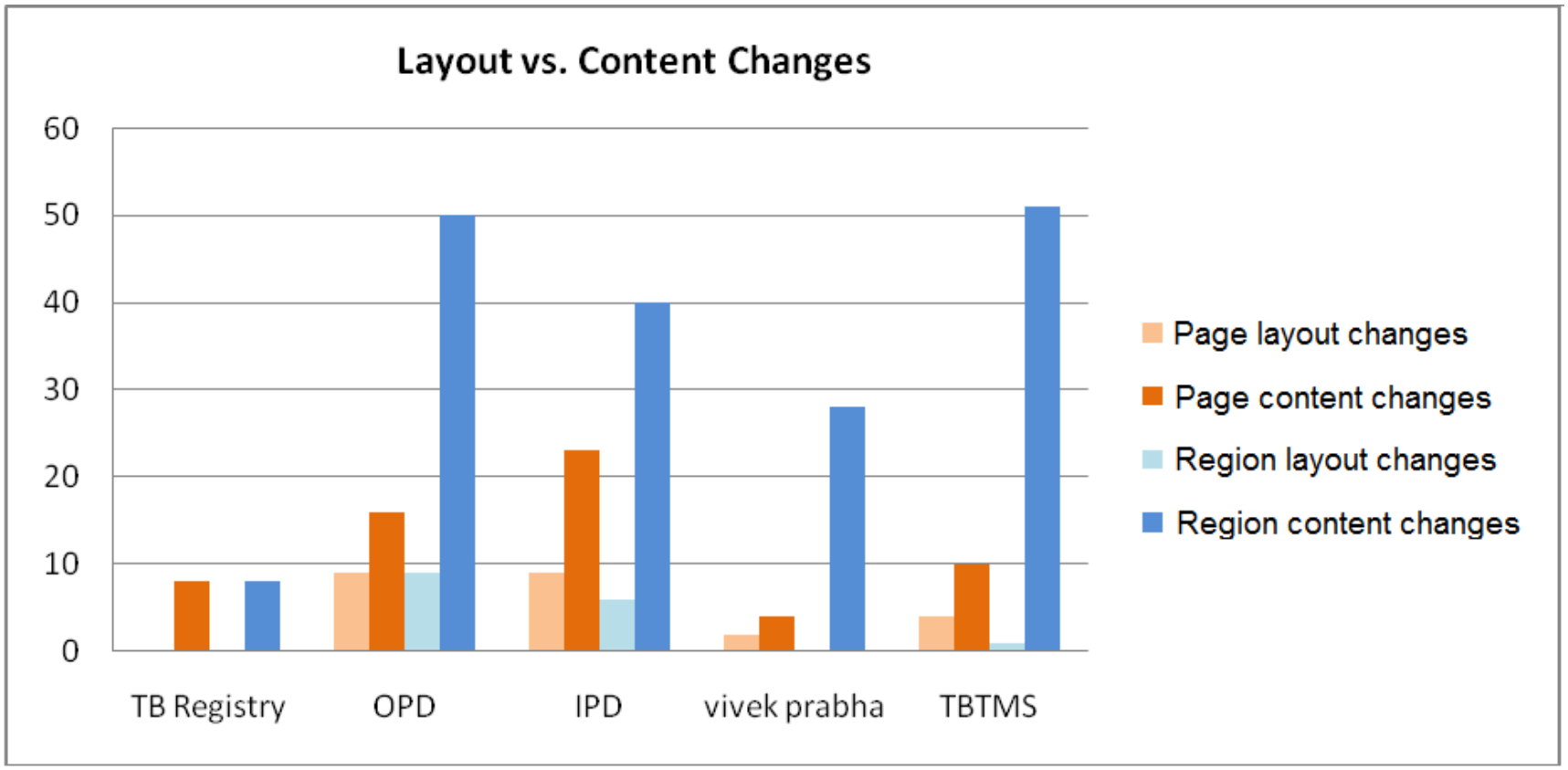
Highest level:

- mostly same, some changes, and few insertions or deletions

Lowest level:

- more insertions and hardly any changes

Evolution Trends



- Content change (darker shades) dominates layout changes.
- Layout changes more at page level than at region level

Outline

- Database Centric Web Applications
  - *Challenges in Application Evolution Tracking*
  - *Ideal Web Application "Diff" Utility*
- Evolution Management Model
  - *Evolution Hierarchies*
  - *Modeling Application Evolution*
- Evolving Application Synopsis Tool
  - *The Oracle APEX Environment*
  - *Tour of EAST*
- Experimental Study
  - *Application Evolution Characteristics*
  - *Experiment Results*
- Conclusion & Future Work

Future Work

- Track evolution of schema objects
  - *Provide a list of application components modified by a change*
- Analyze feasibility of the scheme in other environments
  - *PHP – MySQL*
  - *Ruby on Rails*
  - *ASP.NET – SQL Server*

  where the lack of Metadata in some cases makes this task challenging.

Conclusion

- Presented a scheme for
  - Analyzing evolving database-centric web applications
  - Model the changes.
- Using Model-View-Controller architecture
  - Content changes as well as layout changes.
  - Drill-down capabilities along the hierarchies.
  - Changes with respect to schema dependencies.
- The EAST tool can reduce development and knowledge transition time by automatically generating the application summary and schema dependency “Diffs”.