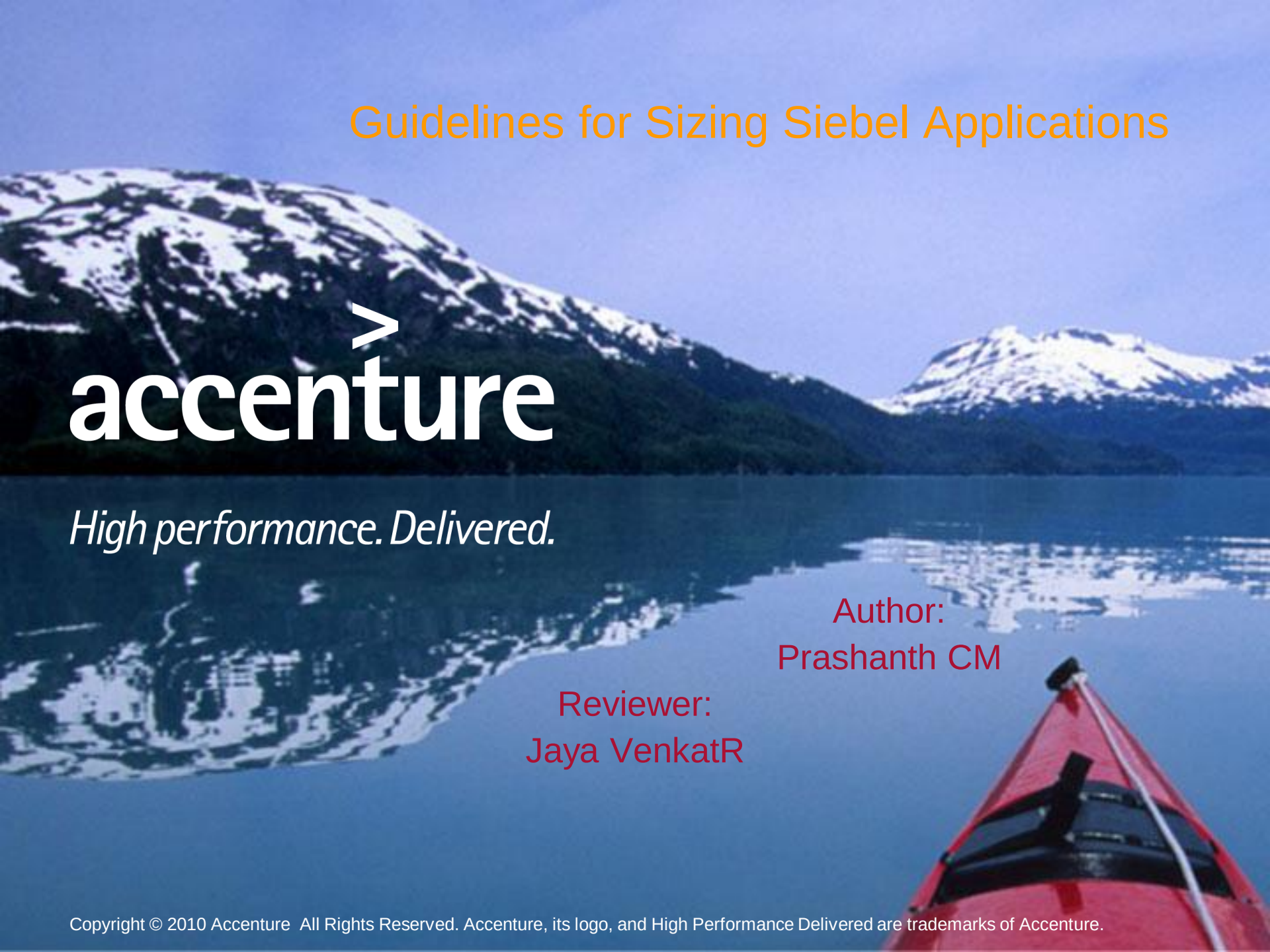
Guidelines for Sizing Siebel Applications



Author:
Prashanth CM

Reviewer:
Jaya VenkatR

Index

Introduction

- Sizing packaged applications has always been a challenge and we always wish to have a defined set of rules and guidelines to follow when there is a need to count package applications.

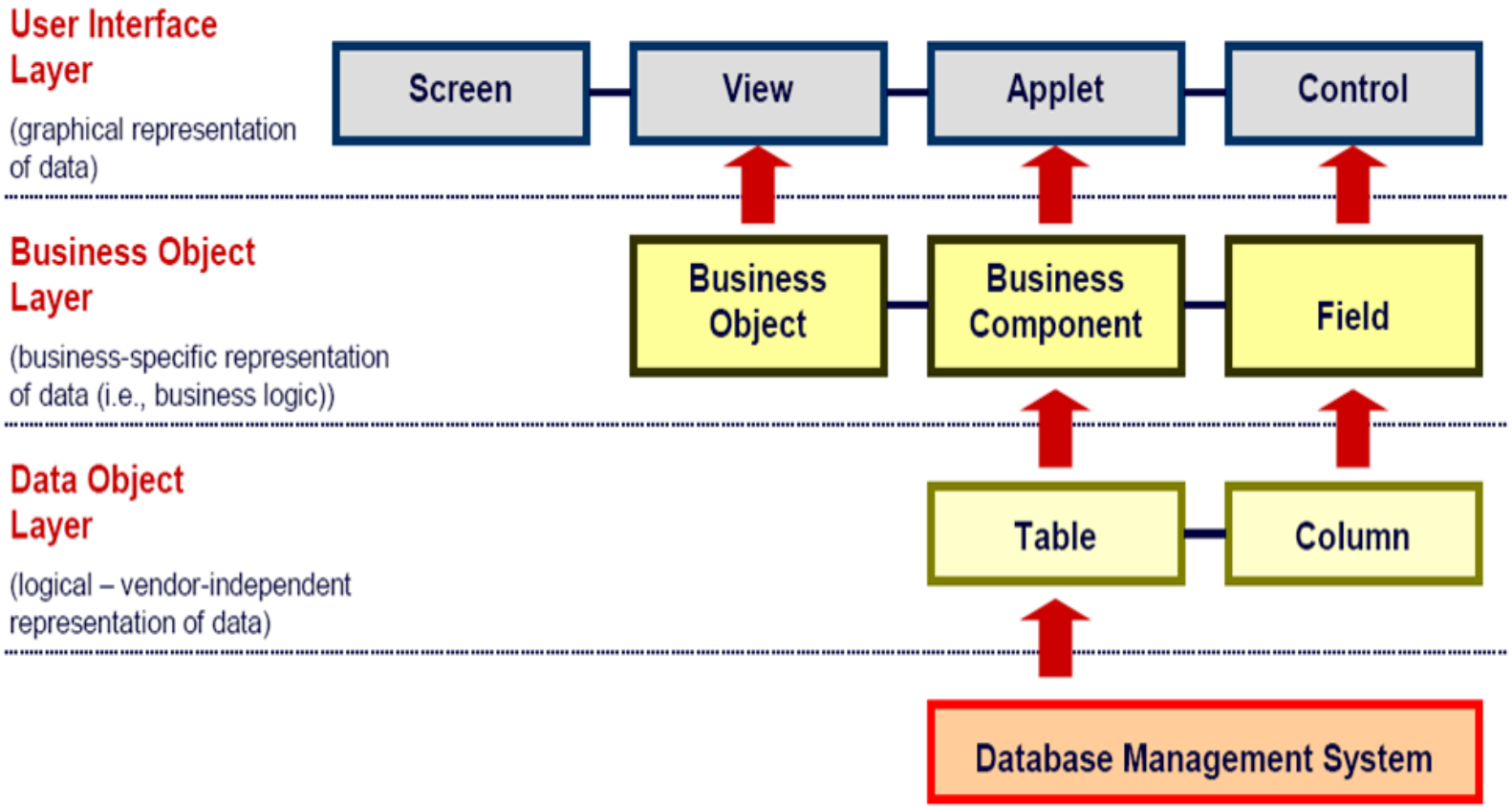
- In this presentation we explore one of the approaches that we can follow when we come across a situation where we have to size Siebel packaged application.

Index

Siebel Architecture

- Siebel's object-based system architecture is designed to provide flexibility concerning the collection and processing information.
- Three different data layers allow an organization to structure information to meet specific business and reporting requirements.

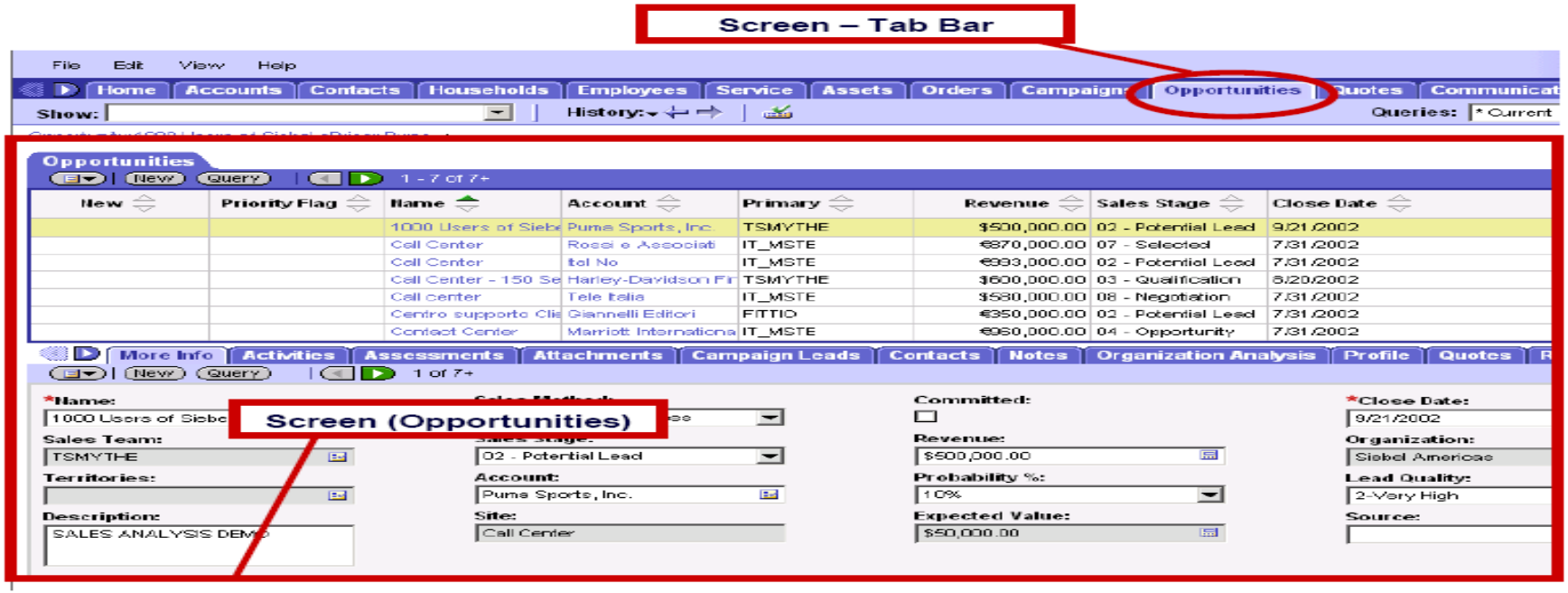
Siebel Object Based Data Model



Siebel User Interface Layer - Screens

Within Siebel, Screens represent groupings of associated tasks that are needed as part of certain job functions.

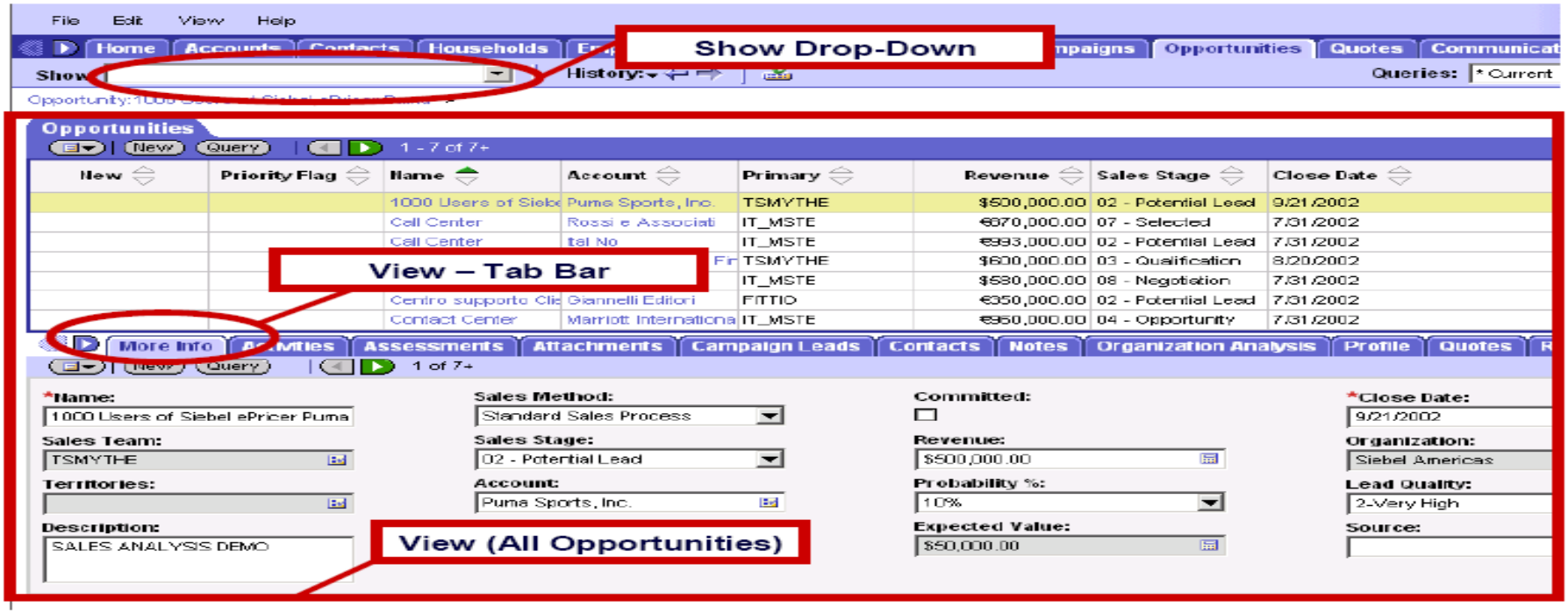
The Opportunities Screen, for instance, is an accumulation of functions, which are needed as part of the opportunity administration/processing.



Siebel User Interface Layer - Views

Siebel uses Views to determine the level of functionality that is accessible to users within a given screen.

Views can also be used to better segregate the access to data by distinguishing the range of data, which is accessible (eg., My Opportunities vs. All Opportunities)

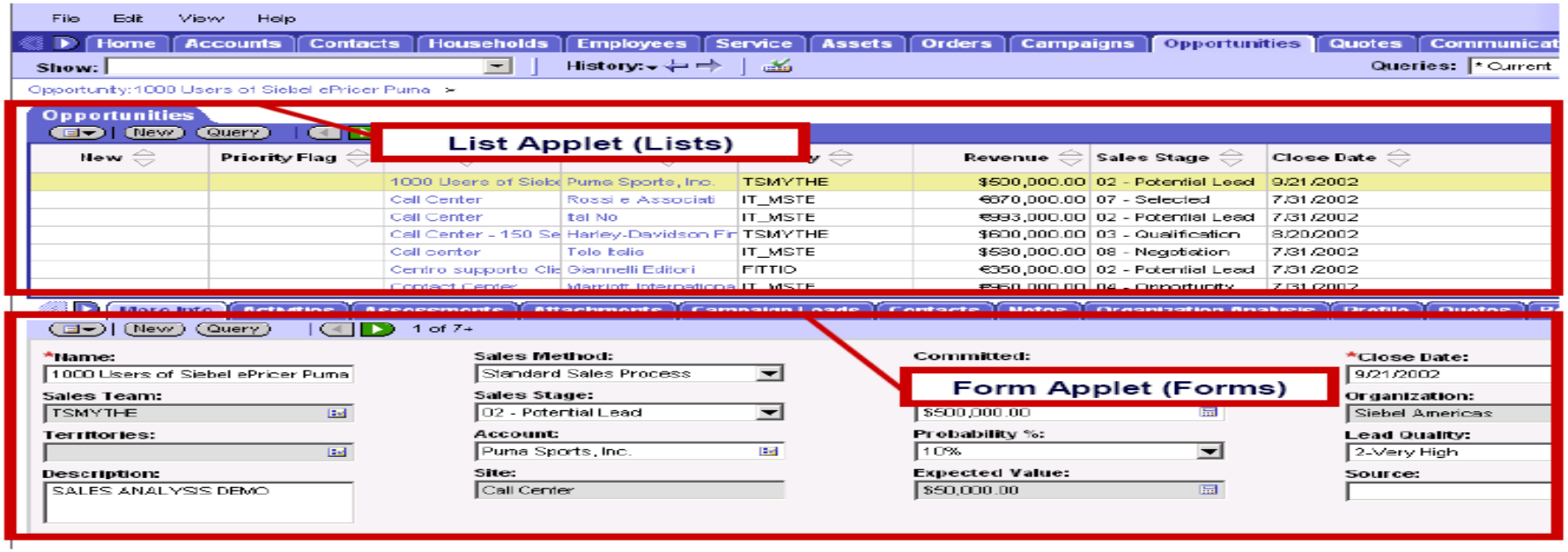


Siebel User Interface Layer - Applets

Applets drive the graphical display of Siebel data.

List Applets are commonly used to provide a list-like appearance of multiple data records.

Form Applets take selected records and conveniently display the relevant data fields next to each other. This format supports data entry and maintenance.



Index

Initial Analysis – Questions to ask

- Is this a third party application?
- Is this a packaged application which has been provided as a solution.
  For example Siebel 8.0 developed by Oracle Siebel for Hospitality solutions
- Has this been customized for current use for the client?
- Is this application being supported by our company?
- Is the support being provided for the complete application or only to the customized part of application?
- What would happen during the next upgrade?
- Will the customized part remain along with the upgrade?
- Have there been any enhancements or will there be any in future for this application?

Initial Analysis – Derive Answers

- Based on the questions asked, we will get answers to FP related questions that we usually do not ask directly (not to confuse the SME) unless he/she has some background knowledge on FP counting.

- For each of the questions asked, we can ask ourselves on FP lines and analyze in steps of what we are going to size.

- We can analyze the application based on below steps

  - Is the application countable – This would be the first check
  - What part of the application is countable (customized part or the whole application) – This would help to draw an application boundary
  - Whether the customized part would remain after the next upgrade – This would help to analyze on what would happen to the size after the upgrade
  - What are the functionalities used by the business? (Tough question for the SME and to Self) – Achievable only with continues interaction with the SME and some analysis to be done together.

Index

Approach

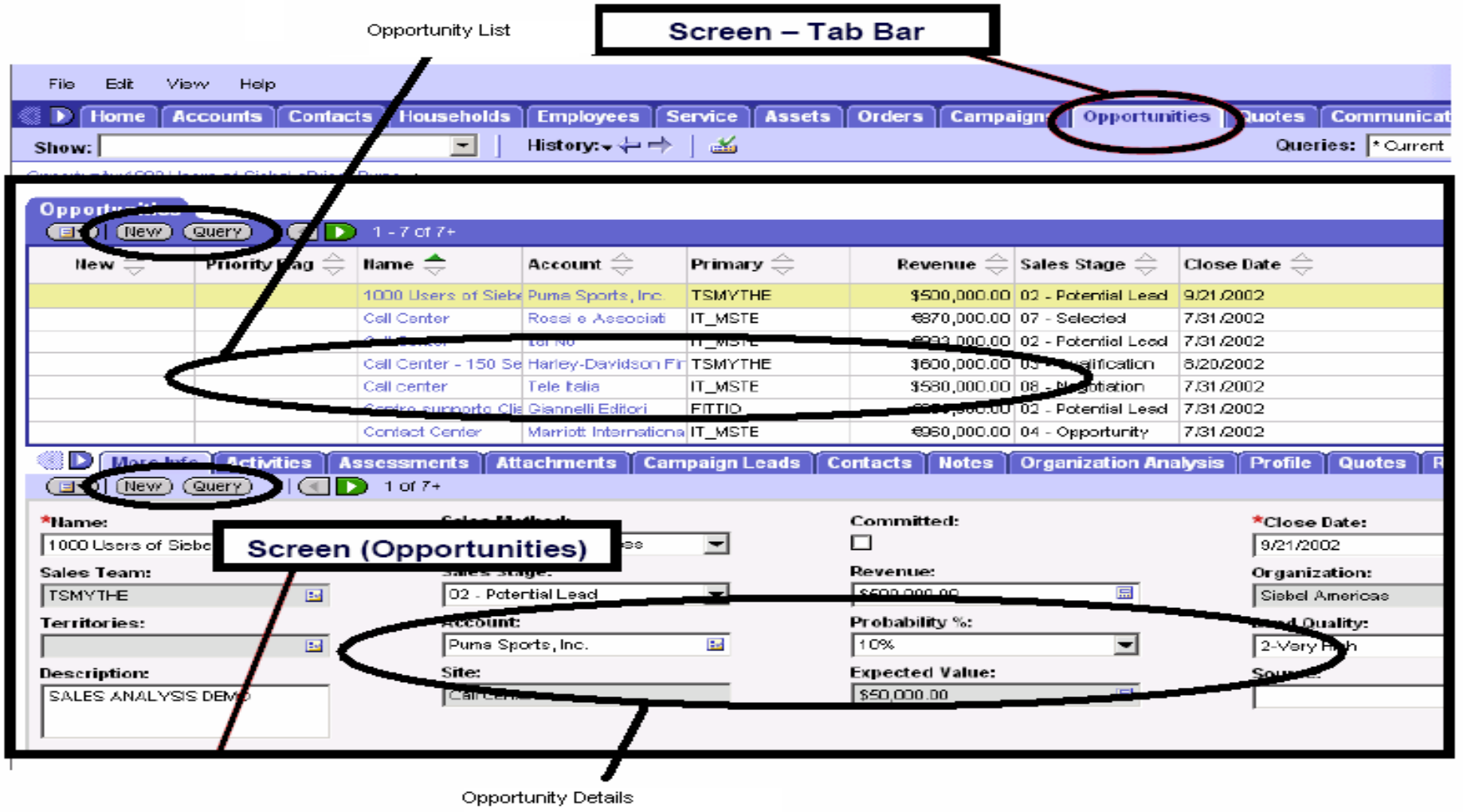
- The Siebel package has more than four thousand tables. A large number of them would not be counted as a logical files.
- To determine the number of actual logical files is a challenge.
- Responsibilities table (S_RESP) describes which user role (RESP_NAME) has the right to use which implemented functionality.
- For example we can get details about the Responsibility Name associated with
  - View
  - Applet
  - Business Object
  - Business Component
- This would give an idea of what areas need to be considered while you count transactions and logical files.

Index

Counting Transactional Functions

- To count transaction functions, best way is to do it through the navigation of screens for which you might require application access.
- Count transactions for each of the business areas.
- Most of the transactions in Siebel would be Create, Update and Deletes apart from analysis charts and other views.
- Best way is to count the transactions through the Site map within the Siebel application where you will find the list of screens that the application uses.
- Browse though the list and count transactions for each screen.

Counting Transactional Functions



Index

Counting Data Functions

Table Naming Conventions in Siebel

Tables in the Siebel database use a three-part naming convention. The syntax is: PREFIX_NAME_SUFFIX.

- PREFIX- Table names in Siebel eBusiness Applications have a one- to three-letter prefix (EIM_, S_, W_, and so on) to distinguish them from other tables in your application.

- NAME- A unique table name that is generally an abbreviation of the entity supertype name.

- SUFFIX- A supertype name may be followed by the entity subtype. For example, the supertype EVT (event) has ACT (activity) as one of its subtypes. Thus, the name becomes S_EVT_ACT.

The prefix indicates the part of the Siebel schema to which a table belongs. Table 1 provides some of the prefixes and their descriptions.

The suffix indicates a table type.
Table 2 provides some of the suffixes and their descriptions.

Counting Data Functions

Table 1. Table Prefixes

Table 2. Base Table Suffixes

Prefix	Meaning
EIM_	Interface tables for Enterprise Integration Manager.
S_	Siebel base table. (Exception: Tables with names of the form S_<name>_IF are obsolete interface tables.)
W_	Siebel eBusiness Data Warehouse table, described in Siebel Data Warehouse Data Model Reference.

Suffix	Meaning
_ATT	File attachment table.
_REL	A table that supports a many-to-many relationship from an entity back to itself.
_SS	A table that stores Siebel-to-Siebel integration information.
_X	One-to-one extension table, available for customers to add attributes to the Siebel database.
_XA	A table that stores extended attributes associated with an object class.
_XM	One-to-many extension table, available for customers to add attributes to the Siebel database.

Counting Data Functions

- Siebel has 3 types of tables namely Data, Interface and Repository
- Data and Repository tables have prefix as “S_” while interface tables have “EIM_” prefix.
- Many data tables have extension tables which have same name as primary data table but they also have a suffix.
- Extension table with 1:1 relationship has suffix “_X”. For e.g. S_CONTACT data table has 1:1 extension table S_CONTACT_X.
- Extension table with 1:M relationship has suffix “_XM”. For e.g. S_CONTACT data table has 1:M extension table S_CONTACT_XM.
- User can add new tables or use extension columns (additional columns provided by Siebel) in existing tables.

Counting Data Functions

- User can also create tables with prefix of “S_”. This can make it difficult to differentiate user’s custom tables from Siebel provided tables. However, for this reason users mostly use a different project specific prefix for their tables.
- Mostly tables with prefix S_ contain business data and need to be considered while counting data functions.
- Have a discussion with the SME to filter the list to be considered for logical grouping.
- For most of the tables the naming convention would be sufficient for you to exclude the entity from counting after deciding that it is a table being used for technical purposes.
- Consider using the Entity Relationship tables from CPM while counting data functions

Index

Conclusion

- For counting Siebel applications, if you are also planning time for SME support, you need to consider adding more time for SME support as the support required would be more than double as per my experience when counting data functions.
- Time spent is more for the analysis that need to be done on the entities unlike counting any other Java based applications where the data model and naming conventions of tables would be more easy to identify.
- One more point to note is the usage of columns in this type of application.
  Some columns might be available in the data structure but might not be used which ideally should not form a part of your count.
- Counting DET's in Siebel Applications remains to be a challenge
- While counting transaction functions care should be taken not to double count functions and the best way is to access the application for transaction functions.

References

- http://download.oracle.com/
- IFPUG CPM 4.2.1
- http://www.isaca-psc.org

accenture