ICFI 2005, Leicester, UK

Feature Interactions Induced by Data Dependencies among Entity Components

Teng TENG, Gang HUANG, Ruichao LI,
Dong ZHAO, Hong MEI

School of Electronics Engineering and Computer Science,
Peking University

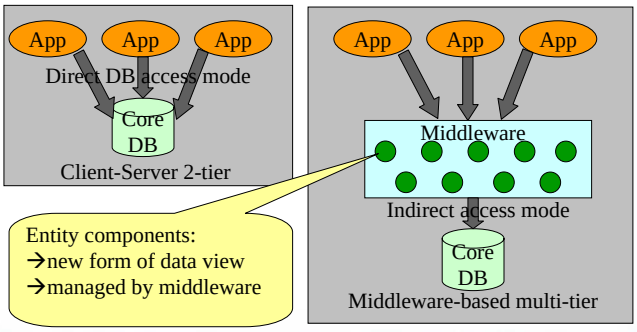
Corresponding: huanggang@sei.pku.edu.cn

Peking University

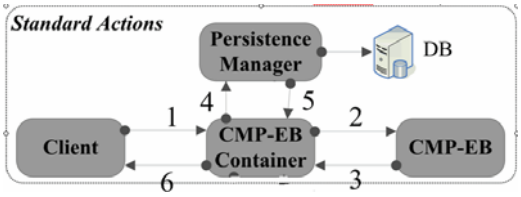
Agenda

- Problem Statement
  - Cases of feature interactions of data dependencies
  - Analysis of the intrinsic reason
- Solution and Implementation
  - Why middleware based
  - Key mechanisms
  - Criteria for Detecting Conflicts
  - Implementation in a J2EE application server
- Conclusion and Future work

From 2-tier to 3-tier

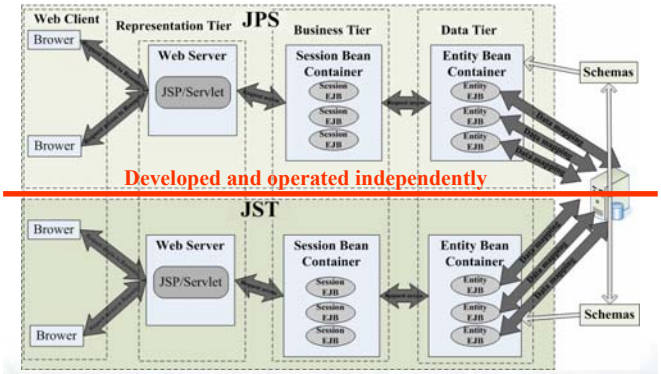


Management of Entity Components



- Existing middleware focuses on individual applications
  - Ignores data dependencies and conflicts when multiple independent applications access the same database
  - Leads to unexpected data operations like feature interactions

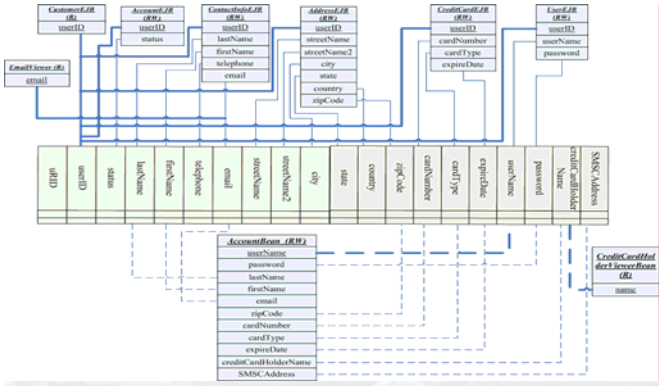
J2EE as Demonstration



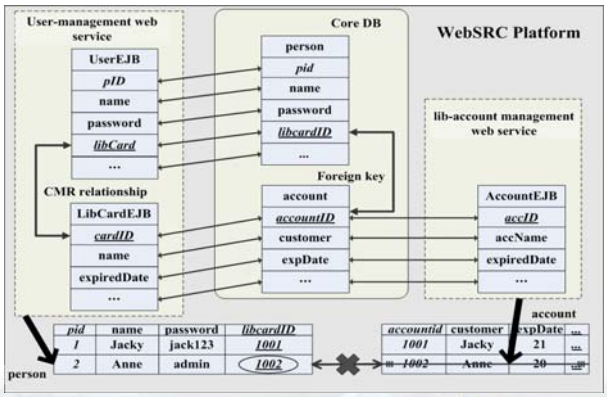
Data Dependencies & Conflicts

- Data Dependency
  - Data-related interactions between subsystems
  - For example, when two subsystems manipulate the same data or when they manipulate the data which have explicit or implicit relationships, they interact with each other in a data-related way.
- Data Conflict
  - When a subsystem fails to manipulate the data in a correct way, other subsystems with data dependencies may not work well and may even crash.
  - Feature interaction problems!!!

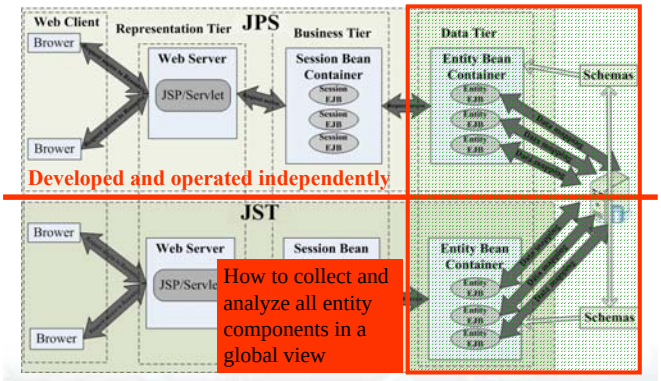
Case 1: Primary Key is NULL



Case 2: Reference Integrity is Violated



The Solution



Agenda

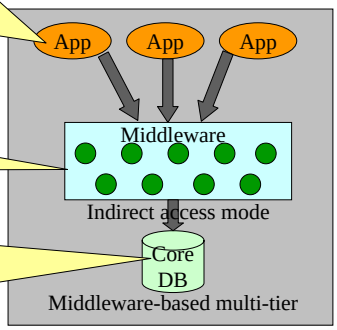
- Problem Statement
  - Cases of feature interactions of data dependencies
  - Analysis of the intrinsic reason
- Solution and Implementation
  - Why middleware based
  - Key mechanisms
  - Criteria for Detecting Conflicts
  - Implementation in a J2EE application server
- Conclusion and Future work

Who can do the Job

For a single application, the knowledge of data information it obtains is completely limited to the scope of its own usage of data. It is not at all aware of the whole data usages of the EIS.

Middleware is the answer

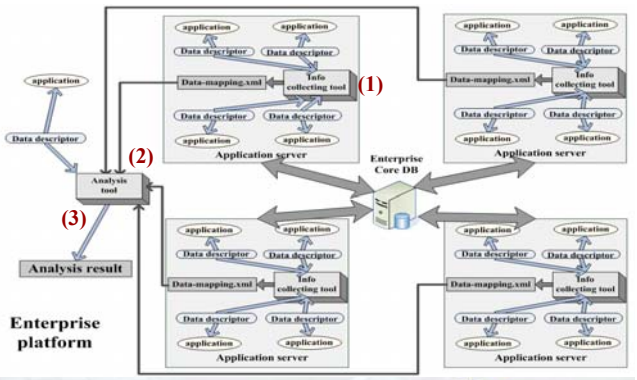
For the core database, it communicates with EIS subsystems using services provided by the middleware which acts as a proxy between the database and the EIS.



Middleware-Based Approach

- Collecting data usage information
  - from deployment descriptors
- Discovering data dependencies
  - from collected data usage information
- Detecting data conflicts
  - criteria for detecting data conflicts
- Developing non-conflict subsystems
  - redesign, redeploy, reconfiguration,...

Middleware-Based Detecting



PKUAS-Based Implementation

- The Steps to develop CMP-EB-based subsystems
  - Provide each CMP-EB according to the EJB specification as an abstract class
  - Write deployment descriptor for each CMP-EB
  - Specify databases, tables and columns arranged for each CMP-EB
- Parsing CMP-EB data deployment descriptors, the data usage collector can collect the required information.
- Analysis tool can discover data dependencies & conflicts based on the information collected.
  - If 2 applications are dependent on the same database table, there exist data dependencies between them.

Data Collection

Table 1. Meanings of the elements in deployment descriptor.

Elements	meanings
entity	declares an entity bean.
ejb-name	specifies an enterprise bean's name.
local-home	contains the fully-qualified name of the bean's local home interface.
local	contains the fully-qualified name of the enterprise bean's local interface.
ejb-class	contains the fully-qualified name of the enterprise bean's class.
persistence-type	specifies an entity bean's persistence management type.
prim-key-class	contains the fully-qualified name of an entity bean's primary key class.
cmp-version	specifies the version of an entity bean with CMP.
cmp-field	describes a container-managed field.
field-name	specifies the name of a container managed field
primkey-field	specify the name of the primary key field for an entity bean with CMP
module	declares an entity bean's deployment descriptor with CMP
name	specifies the name of database which is associated with the entity bean.

A Sample of Deployment Descriptor

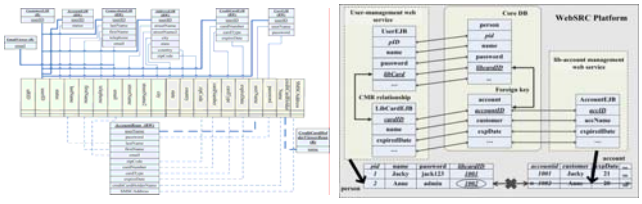
```
<entity>
  <ejb-name>ContactInfoEJB</ejb-name>
  <local-home>
    com.sun.j2ee.blueprints.contactinfo.ejb.ContactInfoLocalHome
  </local-home>
  <local>
    com.sun.j2ee.blueprints.contactinfo.ejb.ContactInfoLocal
  </local>
  <ejb-class>
    com.sun.j2ee.blueprints.contactinfo.ejb.ContactInfoEJB
  </ejb-class>
  <persistence-type>Container</persistence-type>
  <prim-key-class>java.lang.Integer</prim-key-class>
  <cmp-version>2.x</cmp-version>
  <cmp-field>
    <field-name>userID</field-name>
  </cmp-field>
  <cmp-field>
    <field-name>familyName</field-name>
  </cmp-field>
  <cmp-field>
    <field-name>telephone</field-name>
  </cmp-field>
  <cmp-field>
    <field-name>givenName</field-name>
  </cmp-field>
  <cmp-field>
    <field-name>email</field-name>
  </cmp-field>
  <primkey-field>userID</primkey-field>
</entity>
```

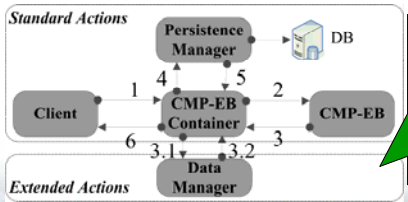
```
<module name="jdbc/oracle" type="javax.sql.DataSource">
  <ejb-name>ContactInfoEJB</ejb-name>
  <ejb-table-name>CustomerTB</ejb-table-name>
  <field-maps>
    <field-map>
      <field-name>userID</field-name>
      <column-name>userID</column-name>
    </field-map>
    <field-map>
      <field-name>givenName</field-name>
      <column-name>lastName</column-name>
    </field-map>
    <field-name>familyName</field-name>
      <column-name>firstName</column-name>
    </field-map>
    <field-map>
      <field-name>telephone</field-name>
      <column-name>tele</column-name>
    </field-map>
    <field-map>
      <field-name>email</field-name>
      <column-name>email</column-name>
    </field-map>
  </field-maps>
</module>
```


Criteria for Detecting Data Conflicts

- If CMP-EB A and CMP-EB B are data dependent, the scenarios of data conflict are in the following:
  - When CMP-EB A and CMP-EB B are both read-only
    - no data conflicts exist
  - When CMP-EB A is read-only and CMP-EB B is writable
    - if the set of the primary key columns of B is a subset of the set which contains the primary key columns of A, and the latter is also a subset of the set including columns associated with B
  - When CMP-EB B is read-only and CMP-EB A is writable
    - if the set of the primary key columns of A is a subset of the set which contains the primary key columns of B, and the latter is also a subset of the set including columns associated with A
  - When CMP-EB A and CMP-EB B are both writable
    - if the set which contains the primary key columns of A is equal to that of B

Results





A Solution:
→assign non-NULL
→prohibit the deletion
→…
It may be incorrect to change the semantics!

Conclusion and Future Work

- **This Paper**
  - Find out that the problem caused by data dependencies and conflicts between entity components is similar to feature interactions in telecom.
  - Demonstrates the approach in an extended J2EE application server to resolve problem caused by data dependencies & conflicts.
- **Future Work**
  - The intrinsic problem is that the middleware only focuses on the raw data and ignores the meta data in the database
  - How to define, instantiate, associate and maintain meta data by middleware?

Peking University