


BEA AquaLogic Data Services Platform

Jose Luis Ambite


[Based on slides by Mike Carey]



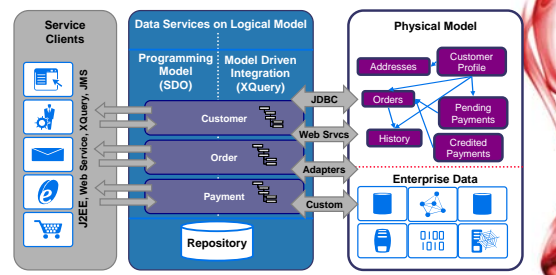
Declarative Integration via XML & XQuery

Requirements	Standards
A standard for data format and data interchange	➔ XML
A standard for describing and modeling data	➔ XML Schema
A standard for interfacing into applications	➔ Web Services
A standard for querying both relational and non-relational data	➔ XQuery
A standard Java programming model (read + write)	➔ SDO (Service Data Objects)
A standard for publishing consumable services	➔ Web Services

© Copyright 2007, BEA Systems, Inc. | 2




Data Services: BEA AquaLogic DSP

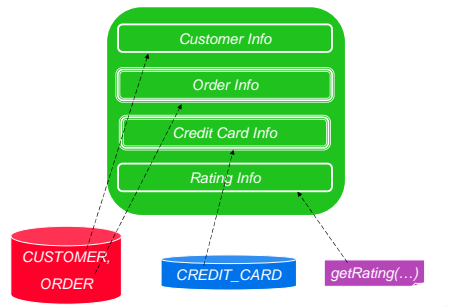


- Data services group operations related to (coarse-grained) **business entities**
- Logical models capture data access and integration complexity **one time**
- **Uniform** data model, programming model, and API for all enterprise data
- Service **clients** may be portals, business processes, composite applications,...


© Copyright 2007, BEA Systems, Inc. | 3



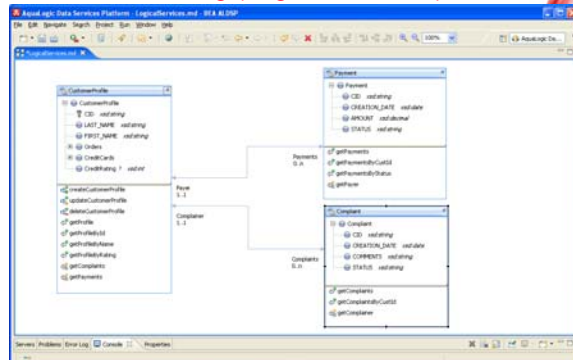
Example: Customer Profile Data Service



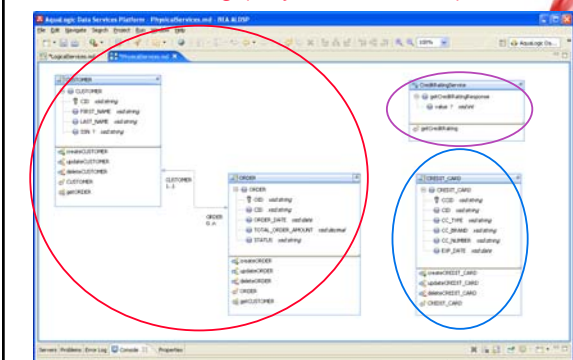
© Copyright 2007, BEA Systems, Inc. | 4



Service Modeling (Logical Services)



Service Modeling (Physical Services)



Data Service – CustomerProfile View

Graphical Service Creation

Data Service – “Get All” Read Method

```

(::pragma function ... kind="read" ...)
declare function tns:getProfile() as element(ns0:PROFILE)*
{
  for $cust in cus:CUSTOMER() return <tns:CustomerProfile>
  <CID>{fn:data($cust/CID)}</CID>
  <LAST_NAME>{fn:data($cust/LAST_NAME)}</LAST_NAME>
  <FIRST_NAME>{fn:data($cust/FIRST_NAME)}</FIRST_NAME>
  <Orders>{
    for $ord in cus:getORDER($cust) return <ORDER>
    <OID>{fn:data($ord/OID)}</OID>
    <CID>{fn:data($ord/CID)}</CID>
    <ORDER_DATE>{fn:data($ord/ORDER_DATE)}</ORDER_DATE>
    <TOTAL>{fn:data($ord/TOTAL_ORDER_AMOUNT)}</TOTAL>
    <STATUS>{fn:data($ord/STATUS)}</STATUS>
  }</Orders>
  <CreditCards>{... (see next slide ...)}</CreditCards>
  <CreditRating>{... (see next slide ...)}</CreditRating>
} </tns:CustomerProfile>
};
  
```

Data Service – “Get All” Read Method (cont.)

```

for $cust in cus:CUSTOMER() return <tns:CustomerProfile>
...
  <CreditCards>{
    for $card in cre:CREDIT_CARD()
    where $cust/CID eq $card/CID return <CREDIT_CARD>
    <CCID>{fn:data($card/CCID)}</CCID>
    <CID>{fn:data($card/CID)}</CID>
    <TYPE>{fn:data($card/CC_TYPE)}</TYPE>
    <BRAND>{fn:data($card/CC_BRAND)}</BRAND>
    <NUMBER>{fn:data($card/CC_NUMBER)}</NUMBER>
    <EXP_DATE>{fn:data($card/EXP_DATE)}</EXP_DATE>
  }</CreditCards>
  <CreditRating?>{
    fn:data(cre3:getCreditRating(
      <cre2:getCreditRating>
      <cre2:lastName?>{fn:data($cust/LAST_NAME)}
      </cre2:lastName>
      <cre2:ssn?>{fn:data($cust/SSN)}</cre2:ssn>
      </cre2:getCreditRating>)/cre2:value)
  }</CreditRating>
} </tns:CustomerProfile>
};
  
```

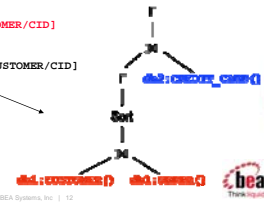
ALDSP Query Processing Overview

Query Processing: Pushdown Example

- Translation to joins and grouping when possible
- Split sorting and grouping

```

for $CUSTOMER in db1:CUSTOMER()
return
  <tns:PROFILE>
  <CID>{ fn:data($CUSTOMER/CID) }</CID>
  ...
  <ORDERS>{
    db1:ORDER()[CID eq $CUSTOMER/CID]
  }</ORDERS>
  <CREDIT_CARDS>{
    db2:CREDIT_CARD()[CID eq $CUSTOMER/CID]
  }</CREDIT_CARDS>
} </tns:PROFILE>
  
```



Query Processing, Example (*getProfile*)

```

1  FLWOR
2  return
3  let $data :=
4  /getRatingFrom
5  <!-- service source: getRating -->
6  <!-- getRating -->
7  <!-- @xmlns="http://www.bea.com/ALDS" -->
8  let $ID :=
9  /getProfile/customer-id="true" profile="true"
10 <!-- @xmlns="http://www.bea.com/ALDS" -->
11 <!-- @xmlns="http://www.bea.com/ALDS" -->
12 <!-- @xmlns="http://www.bea.com/ALDS" -->
13 <!-- @xmlns="http://www.bea.com/ALDS" -->
14 <!-- @xmlns="http://www.bea.com/ALDS" -->
15 <!-- @xmlns="http://www.bea.com/ALDS" -->
16 <!-- @xmlns="http://www.bea.com/ALDS" -->
17 <!-- @xmlns="http://www.bea.com/ALDS" -->
18 <!-- @xmlns="http://www.bea.com/ALDS" -->
19 <!-- @xmlns="http://www.bea.com/ALDS" -->
20 <!-- @xmlns="http://www.bea.com/ALDS" -->
21 <!-- @xmlns="http://www.bea.com/ALDS" -->
22 <!-- @xmlns="http://www.bea.com/ALDS" -->
23 <!-- @xmlns="http://www.bea.com/ALDS" -->
24 <!-- @xmlns="http://www.bea.com/ALDS" -->
25 <!-- @xmlns="http://www.bea.com/ALDS" -->
26 <!-- @xmlns="http://www.bea.com/ALDS" -->
27 <!-- @xmlns="http://www.bea.com/ALDS" -->
28 <!-- @xmlns="http://www.bea.com/ALDS" -->
29 <!-- @xmlns="http://www.bea.com/ALDS" -->
30 <!-- @xmlns="http://www.bea.com/ALDS" -->
31 <!-- @xmlns="http://www.bea.com/ALDS" -->
32 <!-- @xmlns="http://www.bea.com/ALDS" -->
33 <!-- @xmlns="http://www.bea.com/ALDS" -->
34 <!-- @xmlns="http://www.bea.com/ALDS" -->
35 <!-- @xmlns="http://www.bea.com/ALDS" -->
36 <!-- @xmlns="http://www.bea.com/ALDS" -->
37 <!-- @xmlns="http://www.bea.com/ALDS" -->
38 <!-- @xmlns="http://www.bea.com/ALDS" -->
39 <!-- @xmlns="http://www.bea.com/ALDS" -->
40 <!-- @xmlns="http://www.bea.com/ALDS" -->
41 <!-- @xmlns="http://www.bea.com/ALDS" -->
42 <!-- @xmlns="http://www.bea.com/ALDS" -->
43 <!-- @xmlns="http://www.bea.com/ALDS" -->
44 <!-- @xmlns="http://www.bea.com/ALDS" -->
45 <!-- @xmlns="http://www.bea.com/ALDS" -->
46 <!-- @xmlns="http://www.bea.com/ALDS" -->
47 <!-- @xmlns="http://www.bea.com/ALDS" -->
48 <!-- @xmlns="http://www.bea.com/ALDS" -->
49 <!-- @xmlns="http://www.bea.com/ALDS" -->
50 <!-- @xmlns="http://www.bea.com/ALDS" -->
51 <!-- @xmlns="http://www.bea.com/ALDS" -->
52 <!-- @xmlns="http://www.bea.com/ALDS" -->
53 <!-- @xmlns="http://www.bea.com/ALDS" -->
54 <!-- @xmlns="http://www.bea.com/ALDS" -->
55 <!-- @xmlns="http://www.bea.com/ALDS" -->
56 <!-- @xmlns="http://www.bea.com/ALDS" -->
57 <!-- @xmlns="http://www.bea.com/ALDS" -->
58 <!-- @xmlns="http://www.bea.com/ALDS" -->
59 <!-- @xmlns="http://www.bea.com/ALDS" -->
60 <!-- @xmlns="http://www.bea.com/ALDS" -->
61 <!-- @xmlns="http://www.bea.com/ALDS" -->
62 <!-- @xmlns="http://www.bea.com/ALDS" -->
63 <!-- @xmlns="http://www.bea.com/ALDS" -->
64 <!-- @xmlns="http://www.bea.com/ALDS" -->
65 <!-- @xmlns="http://www.bea.com/ALDS" -->
66 <!-- @xmlns="http://www.bea.com/ALDS" -->
67 <!-- @xmlns="http://www.bea.com/ALDS" -->
68 <!-- @xmlns="http://www.bea.com/ALDS" -->
69 <!-- @xmlns="http://www.bea.com/ALDS" -->
70 <!-- @xmlns="http://www.bea.com/ALDS" -->
71 <!-- @xmlns="http://www.bea.com/ALDS" -->
72 <!-- @xmlns="http://www.bea.com/ALDS" -->
73 <!-- @xmlns="http://www.bea.com/ALDS" -->
74 <!-- @xmlns="http://www.bea.com/ALDS" -->
75 <!-- @xmlns="http://www.bea.com/ALDS" -->
76 <!-- @xmlns="http://www.bea.com/ALDS" -->
77 <!-- @xmlns="http://www.bea.com/ALDS" -->
78 <!-- @xmlns="http://www.bea.com/ALDS" -->
79 <!-- @xmlns="http://www.bea.com/ALDS" -->
80 <!-- @xmlns="http://www.bea.com/ALDS" -->
81 <!-- @xmlns="http://www.bea.com/ALDS" -->
82 <!-- @xmlns="http://www.bea.com/ALDS" -->
83 <!-- @xmlns="http://www.bea.com/ALDS" -->
84 <!-- @xmlns="http://www.bea.com/ALDS" -->
85 <!-- @xmlns="http://www.bea.com/ALDS" -->
86 <!-- @xmlns="http://www.bea.com/ALDS" -->
87 <!-- @xmlns="http://www.bea.com/ALDS" -->
88 <!-- @xmlns="http://www.bea.com/ALDS" -->
89 <!-- @xmlns="http://www.bea.com/ALDS" -->
90 <!-- @xmlns="http://www.bea.com/ALDS" -->
91 <!-- @xmlns="http://www.bea.com/ALDS" -->
92 <!-- @xmlns="http://www.bea.com/ALDS" -->
93 <!-- @xmlns="http://www.bea.com/ALDS" -->
94 <!-- @xmlns="http://www.bea.com/ALDS" -->
95 <!-- @xmlns="http://www.bea.com/ALDS" -->
96 <!-- @xmlns="http://www.bea.com/ALDS" -->
97 <!-- @xmlns="http://www.bea.com/ALDS" -->
98 <!-- @xmlns="http://www.bea.com/ALDS" -->
99 <!-- @xmlns="http://www.bea.com/ALDS" -->
100 <!-- @xmlns="http://www.bea.com/ALDS" -->

```

Key Point: It's left up to ALDSP to decide how to optimize and execute the *getProfile* data service call. (Just like w/SQL: just say what, don't say how.)

Query Processing, Example 2 (*query getProfile*)

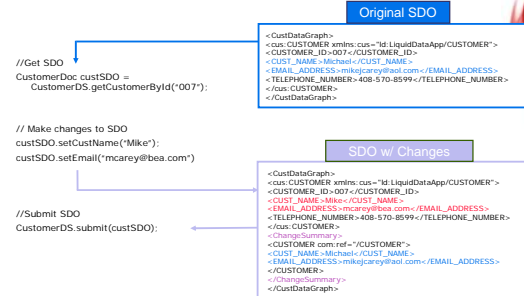
```

1  FLWOR
2  return
3  let $data :=
4  /getRatingFrom
5  <!-- service source: getRating -->
6  <!-- getRating -->
7  <!-- @xmlns="http://www.bea.com/ALDS" -->
8  let $ID :=
9  /getProfile/customer-id="true" profile="true"
10 <!-- @xmlns="http://www.bea.com/ALDS" -->
11 <!-- @xmlns="http://www.bea.com/ALDS" -->
12 <!-- @xmlns="http://www.bea.com/ALDS" -->
13 <!-- @xmlns="http://www.bea.com/ALDS" -->
14 <!-- @xmlns="http://www.bea.com/ALDS" -->
15 <!-- @xmlns="http://www.bea.com/ALDS" -->
16 <!-- @xmlns="http://www.bea.com/ALDS" -->
17 <!-- @xmlns="http://www.bea.com/ALDS" -->
18 <!-- @xmlns="http://www.bea.com/ALDS" -->
19 <!-- @xmlns="http://www.bea.com/ALDS" -->
20 <!-- @xmlns="http://www.bea.com/ALDS" -->
21 <!-- @xmlns="http://www.bea.com/ALDS" -->
22 <!-- @xmlns="http://www.bea.com/ALDS" -->
23 <!-- @xmlns="http://www.bea.com/ALDS" -->
24 <!-- @xmlns="http://www.bea.com/ALDS" -->
25 <!-- @xmlns="http://www.bea.com/ALDS" -->
26 <!-- @xmlns="http://www.bea.com/ALDS" -->
27 <!-- @xmlns="http://www.bea.com/ALDS" -->
28 <!-- @xmlns="http://www.bea.com/ALDS" -->
29 <!-- @xmlns="http://www.bea.com/ALDS" -->
30 <!-- @xmlns="http://www.bea.com/ALDS" -->
31 <!-- @xmlns="http://www.bea.com/ALDS" -->
32 <!-- @xmlns="http://www.bea.com/ALDS" -->
33 <!-- @xmlns="http://www.bea.com/ALDS" -->
34 <!-- @xmlns="http://www.bea.com/ALDS" -->
35 <!-- @xmlns="http://www.bea.com/ALDS" -->
36 <!-- @xmlns="http://www.bea.com/ALDS" -->
37 <!-- @xmlns="http://www.bea.com/ALDS" -->
38 <!-- @xmlns="http://www.bea.com/ALDS" -->
39 <!-- @xmlns="http://www.bea.com/ALDS" -->
40 <!-- @xmlns="http://www.bea.com/ALDS" -->
41 <!-- @xmlns="http://www.bea.com/ALDS" -->
42 <!-- @xmlns="http://www.bea.com/ALDS" -->
43 <!-- @xmlns="http://www.bea.com/ALDS" -->
44 <!-- @xmlns="http://www.bea.com/ALDS" -->
45 <!-- @xmlns="http://www.bea.com/ALDS" -->
46 <!-- @xmlns="http://www.bea.com/ALDS" -->
47 <!-- @xmlns="http://www.bea.com/ALDS" -->
48 <!-- @xmlns="http://www.bea.com/ALDS" -->
49 <!-- @xmlns="http://www.bea.com/ALDS" -->
50 <!-- @xmlns="http://www.bea.com/ALDS" -->
51 <!-- @xmlns="http://www.bea.com/ALDS" -->
52 <!-- @xmlns="http://www.bea.com/ALDS" -->
53 <!-- @xmlns="http://www.bea.com/ALDS" -->
54 <!-- @xmlns="http://www.bea.com/ALDS" -->
55 <!-- @xmlns="http://www.bea.com/ALDS" -->
56 <!-- @xmlns="http://www.bea.com/ALDS" -->
57 <!-- @xmlns="http://www.bea.com/ALDS" -->
58 <!-- @xmlns="http://www.bea.com/ALDS" -->
59 <!-- @xmlns="http://www.bea.com/ALDS" -->
60 <!-- @xmlns="http://www.bea.com/ALDS" -->
61 <!-- @xmlns="http://www.bea.com/ALDS" -->
62 <!-- @xmlns="http://www.bea.com/ALDS" -->
63 <!-- @xmlns="http://www.bea.com/ALDS" -->
64 <!-- @xmlns="http://www.bea.com/ALDS" -->
65 <!-- @xmlns="http://www.bea.com/ALDS" -->
66 <!-- @xmlns="http://www.bea.com/ALDS" -->
67 <!-- @xmlns="http://www.bea.com/ALDS" -->
68 <!-- @xmlns="http://www.bea.com/ALDS" -->
69 <!-- @xmlns="http://www.bea.com/ALDS" -->
70 <!-- @xmlns="http://www.bea.com/ALDS" -->
71 <!-- @xmlns="http://www.bea.com/ALDS" -->
72 <!-- @xmlns="http://www.bea.com/ALDS" -->
73 <!-- @xmlns="http://www.bea.com/ALDS" -->
74 <!-- @xmlns="http://www.bea.com/ALDS" -->
75 <!-- @xmlns="http://www.bea.com/ALDS" -->
76 <!-- @xmlns="http://www.bea.com/ALDS" -->
77 <!-- @xmlns="http://www.bea.com/ALDS" -->
78 <!-- @xmlns="http://www.bea.com/ALDS" -->
79 <!-- @xmlns="http://www.bea.com/ALDS" -->
80 <!-- @xmlns="http://www.bea.com/ALDS" -->
81 <!-- @xmlns="http://www.bea.com/ALDS" -->
82 <!-- @xmlns="http://www.bea.com/ALDS" -->
83 <!-- @xmlns="http://www.bea.com/ALDS" -->
84 <!-- @xmlns="http://www.bea.com/ALDS" -->
85 <!-- @xmlns="http://www.bea.com/ALDS" -->
86 <!-- @xmlns="http://www.bea.com/ALDS" -->
87 <!-- @xmlns="http://www.bea.com/ALDS" -->
88 <!-- @xmlns="http://www.bea.com/ALDS" -->
89 <!-- @xmlns="http://www.bea.com/ALDS" -->
90 <!-- @xmlns="http://www.bea.com/ALDS" -->
91 <!-- @xmlns="http://www.bea.com/ALDS" -->
92 <!-- @xmlns="http://www.bea.com/ALDS" -->
93 <!-- @xmlns="http://www.bea.com/ALDS" -->
94 <!-- @xmlns="http://www.bea.com/ALDS" -->
95 <!-- @xmlns="http://www.bea.com/ALDS" -->
96 <!-- @xmlns="http://www.bea.com/ALDS" -->
97 <!-- @xmlns="http://www.bea.com/ALDS" -->
98 <!-- @xmlns="http://www.bea.com/ALDS" -->
99 <!-- @xmlns="http://www.bea.com/ALDS" -->
100 <!-- @xmlns="http://www.bea.com/ALDS" -->

```

Key Point: Declarative data services, such as those built with ALDSP, support the *performant* reuse of data services.

Updates: Service Data Objects (SDO)



ALDSP inspired by *MultiBase* (Early 1980's)

- One of the first DBMS projects to relax the homogeneity assumption (vs. Ingres, R, ...)
- Interesting foundation and technical contributions
 - ▶ Functional data model (i.e., "everything is a function")
 - Prehistoric objects with identity
 - Functions model attribute access, relationship navigation
 - Model realized via the DAPLEX query language
 - ▶ Important technical achievements
 - Function-based model to normalize relational, network, and other models
 - Early results on federated query processing, sets/multisets, and more
 - Note: This is why computer scientist students should study "history"....
- Functions everywhere --> Web services

Summary of Information Integration

- Components of Information Integration
 - ▶ Understand sources, terminologies
 - ▶ Cleanse, Record Linkage
 - ▶ Define target/mediated Schema
- Approaches to Information Integration
 - ▶ ETL workflows
 - ▶ Virtual Integration
- Representation/Query language
 - ▶ Relational
 - ▶ XML/XQuery/Web Services