Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (0223) 323010**
INTERNATIONAL: **+44 223 323010**
FAX: **+44 223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

Remote Database Queries and ODP (slides)

Gomer Thomas

Abstract

Both the remote database access (RDA) paradigm, based on remote query language access to data, and the remote procedure call (RPC) paradigm, based on remote calls to predefined procedures, have become increasingly important in recent years. Each has unique advantages and disadvantages. Unfortunately, they are currently supported by different protocols and different client-server infrastructures, which makes it very difficult in practice to take advantage of their combined capabilities. This talk describes how the object based concepts of the ANSA/ODP computational model for distributed computing can be used to align these two paradigms, thereby supporting mixed paradigm programming. The model is applicable to current generation relational databases and SQL, and it lays a solid foundation for future object databases and object query languages.

APM.1160.00.01 **Draft** 24 February 1994

Request for Comments (confidential to ANSA consortium for 2 years)

Distribution:
Supersedes:
Superseded by:

Remote Database Queries and ODP (slides)

Remote Database Queries and ODP (slides)

Gomer Thomas

APM.1160.00.01

24 February 1994

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by Architecture Projects Management Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

Architecture Projects Management Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK	(0223) 323010
INTERNATIONAL	+44 223 323010
FAX	+44 223 359779
E-MAIL	apm@ansa.co.uk

REMOTE DATABASE ACCESS
AND
DISTRIBUTED COMPUTING

GOMER THOMAS
Bellcore secondee to ANSA

WHAT'S THE PROBLEM?

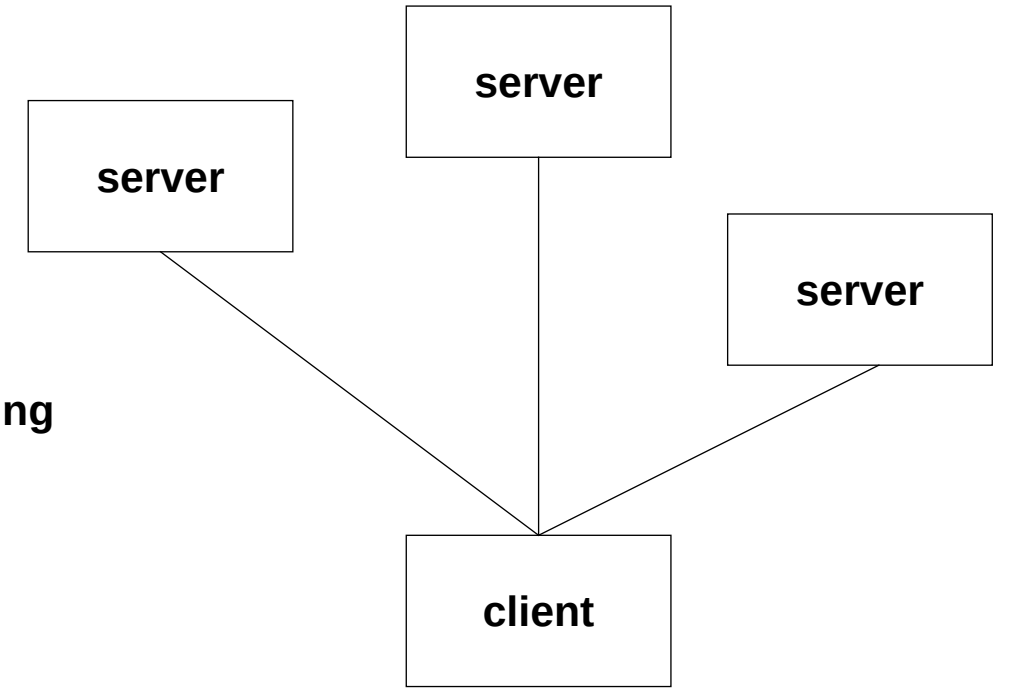
SQL-based remote database access
vs
RPC-based distributed computing

- **Each has unique advantages**
- **How can applications take advantage of both?**

NOTE: The object interaction model is essentially an RPC model, with a powerful framework for structuring the RPC calls

RDA PARADIGM

- **Basic Features**
  - **Transparent remote SQL**
  - **Dialog interaction**
  - **Dynamic SQL**
- **Enhancements**
  - **Distributed transaction processing**
  - **Distributed query processing**
  - **Static SQL**
  - **Stored procedures**



RDA ADVANTAGES

- **Flexibility for clients to define application specific views and high level operations**
- **Vendor-supported interfaces to application development tools**
  - **data browsers**
  - **report writers**
  - **4GLs**
  - **spread sheets**
  - **graphing packages**
  - **etc.**

RPC PARADIGM

- **Basic features**
  - **transparent remote procedures**
  - **“one-shot” interaction**
  - **static procedures**
-
- **Enhancements**
  - **“transactional RPC” (dialog interaction)**
  - **distributed transaction processing**
  - **dynamic procedure definition**
  - **object-orientation**

RPC ADVANTAGES

- **High level of abstraction**
  - **client application programmer convenience**
  - **semantic integrity protection**
  - **low communications traffic**
- **Static compilation**
  - **efficiency**
  - **automated type checking**
- **Applicability to non-database services**

CONCEPTUAL MODEL: DATABASE

- **Entity objects**
- **Set objects**
- **Database objects**

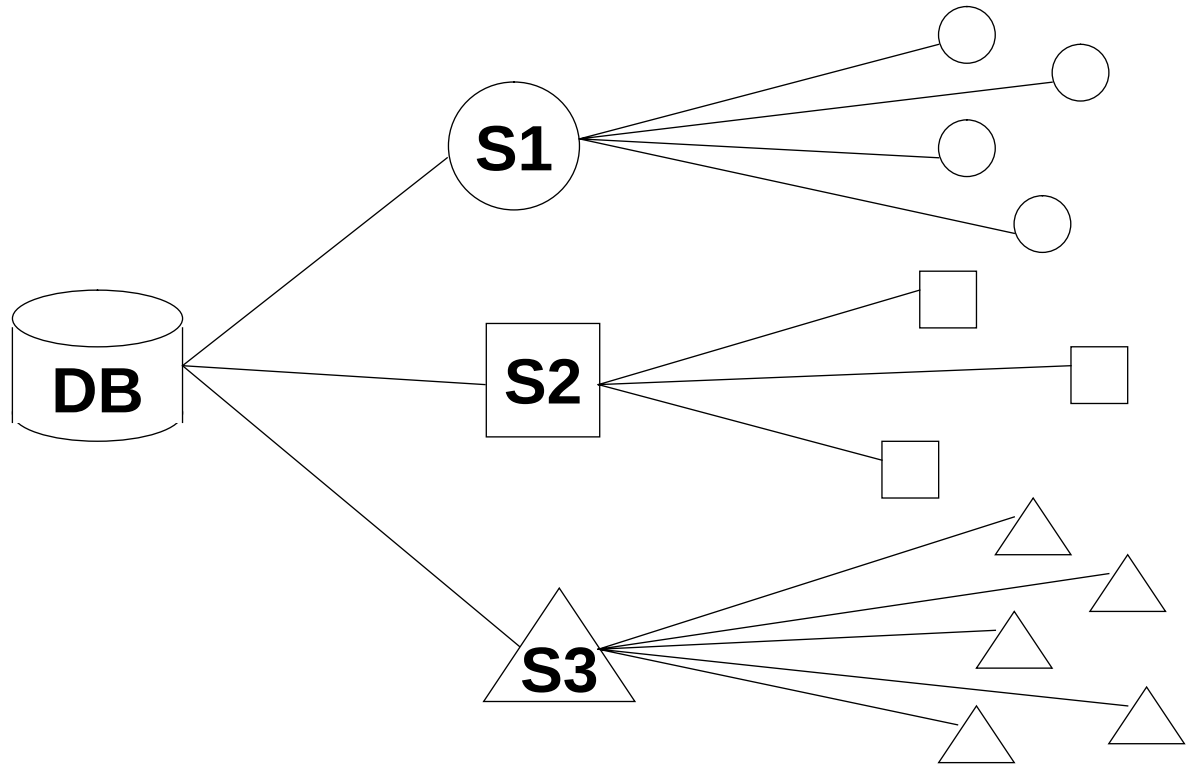
Set object is collection of references to entity objects, all of same type.
Database object is collection of references to set objects.

Each level has its own operations, which invoke operations on lower levels as needed.

No assumption is made about locations of any of the objects.

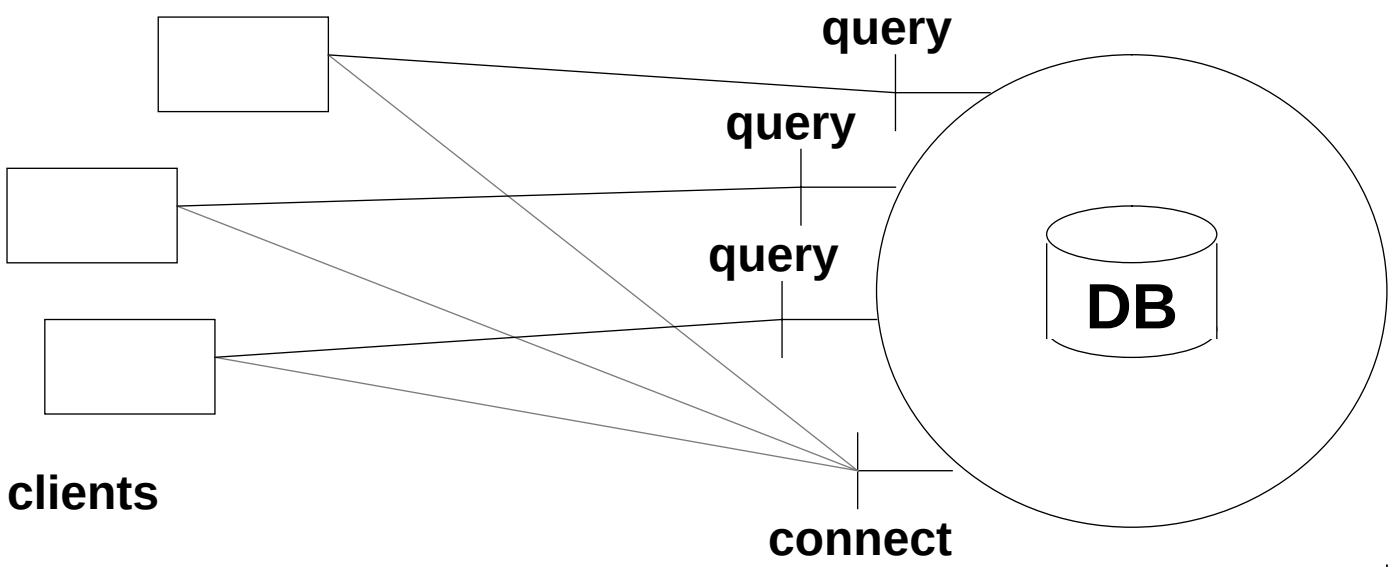
Note: In practice transactions usually require database, or at least set, operations. Even a transaction which only updates a single object must usually first invoke a set operation to select the object to be updated, since the object is typically identified by values of certain attributes, rather than by object id or reference.

CONCEPTUAL MODEL: DATABASE



CONCEPTUAL MODEL: CLIENT-SERVER INTERACTION

- “Connection” interface (obtained by client from trader)
- “Query” Interfaces (set up for individual clients on connection request)



CONCEPTUAL MODEL: RETRIEVALS

- **Query statement is operation constructor**
  - **defines operation type (signature), given schema**
  - **defines result value, given DB state**
- **Step 1: Invoke selection to generate result**
  - **input is query string and parameters appearing in query string, if any**
  - **output is set of entities**
- **Step 2: Invoke operations on result (typically including fetches)**
  - **input to fetch is reference to set of entities**
  - **output of fetch is (reference to) individual entity from set**

CONCEPTUAL MODEL: UPDATES/DELETES

- **Searched update/delete:**

  Query statement is operation constructor
  - **input is query string and parameters appearing in query string, if any**
  - **no output (except possibly status code)**

- **Positioned update/delete:**

  Treat as operations on fetched object

CONCEPTUAL MODEL: INSERTS

- **Single object insert:**

  straightforward -- create object, include reference in set

- **Computed set insert:**

  Step 1: Generate result set to be inserted, via select

  Step 2: Merge into target set

CONCEPTUAL MODEL: QUERIES IN GENERAL

- **In all cases one wants to view a query statement as a constructor, which dynamically defines an operation**

- **The RPC (or object operation invocation) infrastructure needs to support this sort of dynamic operation definition**

TYPE CHECKING

- **Problem:** **Type of operation is dynamically defined by query statement (in context of database schema).**
- **Solution:** **Database supports “describe” operation**
  Input = query statement
  Output = type descriptor

Note: The “describe” operation can be used to check that client and server have a common understanding of the parameter types in the messages between them. It cannot usually be used to check correct type matching between program variables and parameters in the messages, since type information on program variables is usually not available at run time.

TRADING FOR QUERY INTERFACES

Sample Possibilities

- **Trader = data dictionary**
  - **data models (with descriptions of semantics)**
  - **mappings to schemas**
  - **references to “connect” interfaces**

- **Minimal trader**
  - **database names**
  - **references to “connect” interfaces**

DISTRIBUTED or FEDERATED DATABASE

- **Consists of collection of references to databases, together with appropriate cross-database operations (which invoke operations on component databases as needed)**
- **Database components of distributed or federated database may themselves be distributed or federated databases.**

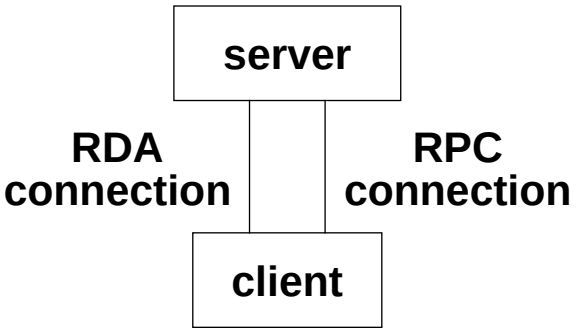
Note that the key difference between an ordinary database and a distributed database in this model is not how the actual data is distributed, but how the functionality is distributed.

IN THE MEANTIME ... (1)

Partial RDA/RPC Integration

- **Stored procedures**
  - **not suitable for non-database processes or services**
  - **not standardised (so far)**

- **RDA + RPC in tandem**
  - **difficult to coordinate RDA thread and RPC thread from same client application at same server**



IN THE MEANTIME ... (2)

RDA over RPC

- **RDA over RPC as transport protocol**
  - **Short term advantage: easy to adapt current products**
  - **Long term disadvantage: awkward and not well adapted to future object databases**
- **RDA over dynamic RPC (as suggested by model presented here)**
  - **Architecturally “clean” and well adapted to future object databases**
- **Hybrid (e.g., map X/Open CLI API to RPC)**
  - **All the disadvantages and none of the advantages of the above two approaches**

SUMMARY

- **Object based conceptual model for remote query language access**
- **Elegant unification of RDA and RPC paradigms**
- **Natural interpretation in context of relational databases and SQL**
- **Solid foundation for future object databases**