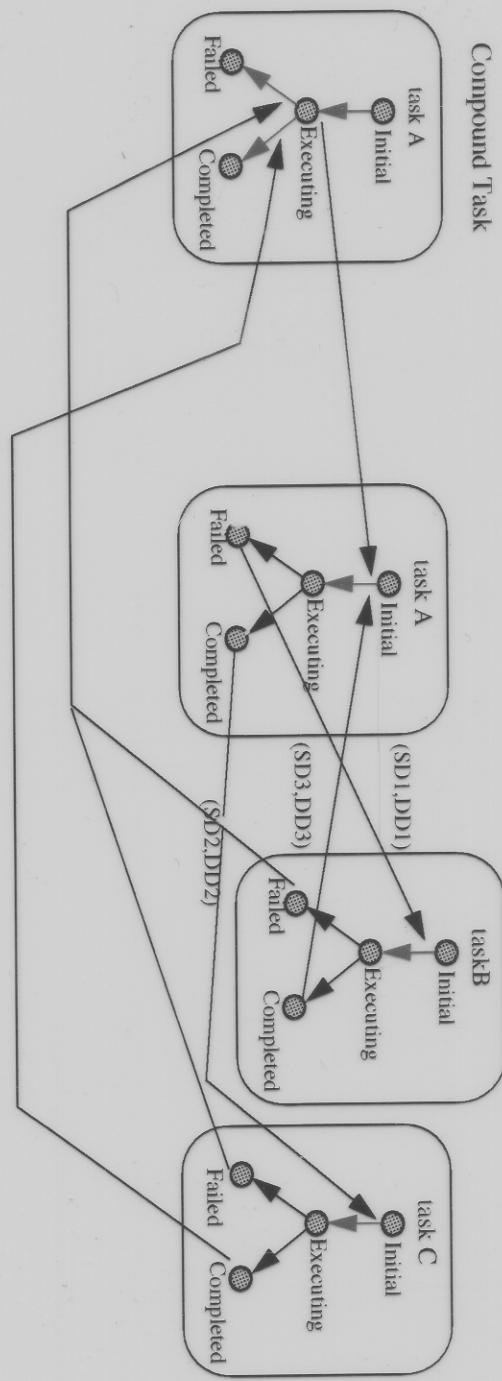
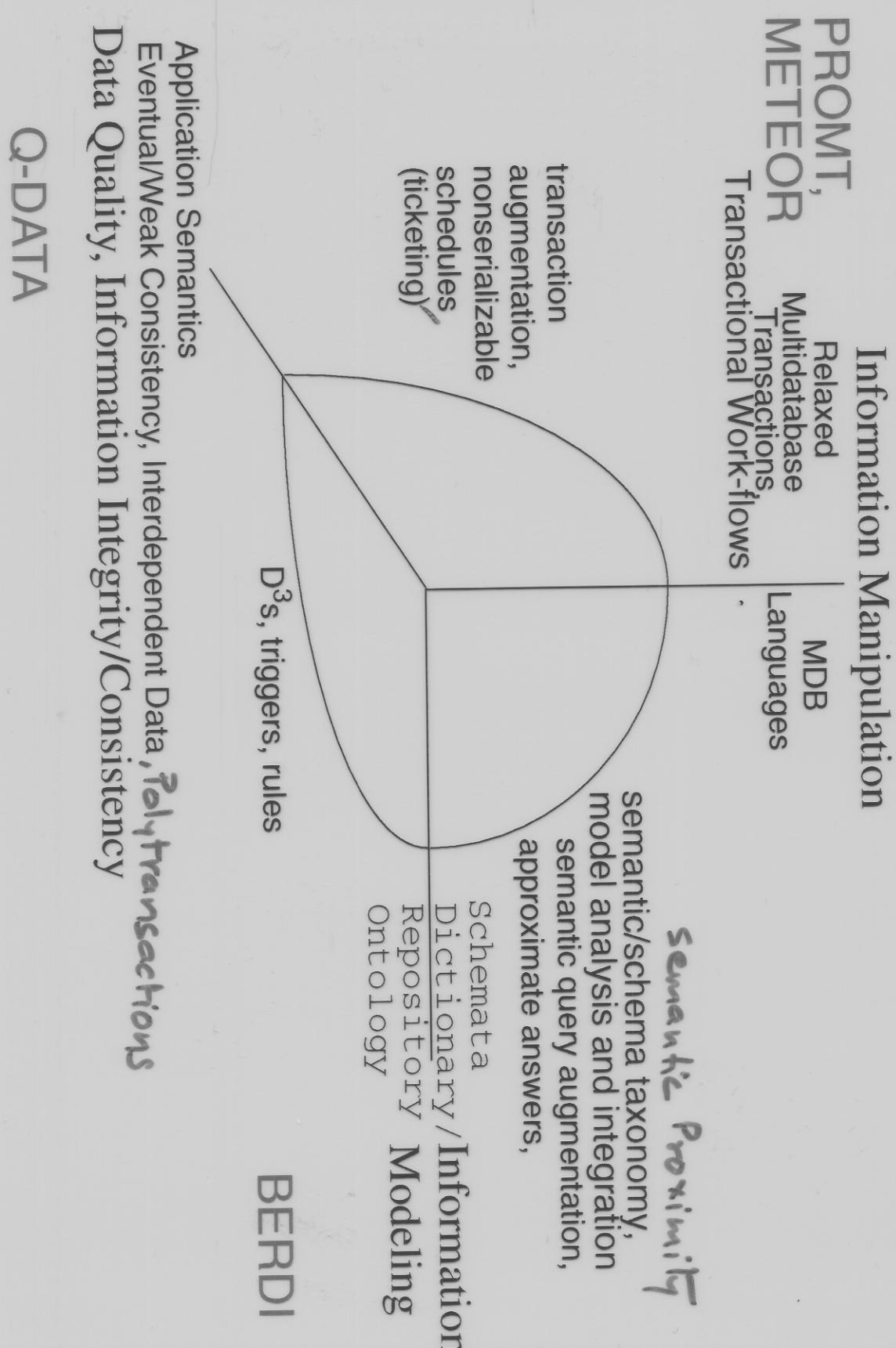


Simple Workflow Example



Issues in Interoperability in Information Systems



Previous Research

- Distributed Databases: heterogeneous distributed DBMS (2LQ), network adaptation, concurrency control, recovery
- Distributed and Parallel Systems: fault tolerance, database machines, parallel logic-database processing (VLPDF)
- Factory Information System Design (Automated Airframe Assembly Program)
- Federated databases: architecture, tools (schema integration, view update- TAILOR), dictionary
- AI-DBMS Integration (*BraID*)

Research Research concerns:

- Information Modeling deals with conceptual level aspects of the workflow, schema and message exchange with an intent to facilitate reuse.
- in automating integration process*
- semantic and schematic taxonomy, context representation
- Information At Bellcore, system engineers deal with the conceptual workflow specification, and developers encode the tasks.
- multidatabase constraints - data dependency handling. System errors can be handled at the task level and logical errors at the workflow
- Data cleanup/purification (Q-DATA)
- Multi-system Application interfaces need to accommodate newer (PROMTS, QoS) only the conceptual specification needs to be
- Advanced communication software and applications, electronic market place, *Infocosm*
- controlling cooperative agents
- information brokering
- integration platform for heterogeneous information (InfoHarness)

Example Transactional Task in TSL

```
Database_task (Sp_rec)
SPECIAL_REC Sp_rec;
{
  EXEC SQL INCLUDE SQLCA ;
  EXEC SQL BEGIN DECLARE SECTION;
    int info ;
  EXEC SQL END DECLARE SECTION;
  EXEC SQL WHENEVER SQLERROR goto Failed ;
  TASK_EXECUTING();
  info = extract_info_from_rec(Sp_rec);
  EXEC SQL INSERT INTO INFO_table VALUES (:info) ;
  EXEC SQL COMMIT ;
  TASK_COMMIT();
Failed :
  EXEC SQL ROLLBACK ;
  TASK_ABORT();
}
```

n-transactional Task in TSL

*(
ng and elevation of system error to logical error)*

```
call up to MAX_TRIES times if system failure*/  
>IF_id1,FCIF_id2);  
  
AILURE || QMS_OSS_DOWN_FAILURE )
```

About the environment

Types of Interfaces

- RPC and t-RPC mechanisms using transaction processing systems
- queue managers
- proprietary workstation to mainframe interfaces for
 - contracts
 - terminal emulation

About the environment *Additional Issues*

- Inter-task dependencies
 - state-based
 - value-based: I/O objects and external variables
- Data Management
 - different data formats for input/output (e.g., FCIF)
 - use of auxiliary systems for complex data manipulation (e.g., TVO)

About the environment

Additional Issues (continued)

- Long running? (may be, may be not)
- Error Handling
 - Systems Errors
 - Logical Errors
- Forward Recovery
- Dynamic Aspects
 - processing entity not known at design time
 - new tasks can be added dynamically
 - multiple concurrent invocation of the same task types
 - ...

Technology & Markets

- Workflow Automation Software
 - Ad-hoc
 - Administrative
 - Production
- Transaction Processing Systems
- Multidatabase Transactions

Workflow Management

Three Components:

- Specification: Model and Language
 - specification of tasks,
 - dependencies and data exchange, and
 - intra- and inter-workflow execution requirements (failure atomicity execution atomicity, workflow recovery, inter-workflow concurrency)
- Scheduling:
 - safe, correct, optimal/efficient, failure handling; exploit task and system semantics
- Executing:
 - manage execution of tasks/transactions on heterogeneous, autonomous component systems

Components of WFSL (partial)

- Task types: task structures, data input/output
- Task classes, Task instances
- Inter-task dependencies (logical error handling)
- Data exchange statements
- Filter (interface def.)

Component of TSL (partial)

- processing entity specific statements
- statements for revealing task structures
- statements for identifying interfaces and dealing with systems errors

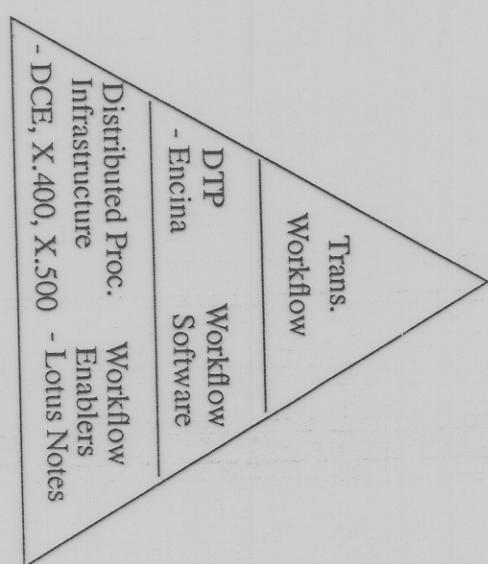
Correctness & Execution Requirements

- Executable/ run-time enforceable specification
- Schedulings
 - correctness (no violation of intertask dependencies)
 - safety (acceptable states)
 - termination (of scheduler)
- Recovery
 - forward recovery
- Concurrency Control
 - serializability ??

Conclusions (continued)

- Better relevance and wider applicability than heterogeneous DDBMS and multidatabase transaction
- It is the way to provide glue to handle legacy systems to support migration/evolution....

Technology Perspective



Projects Developing Prototypes and Addressing Workflow and Related Issues

(a partial list)

Carnot [MCC]

DOM [GTE Lab]

Interbase [Purdue]

Pegasus [HP Lab]

METEOR [Bellcore, ...]