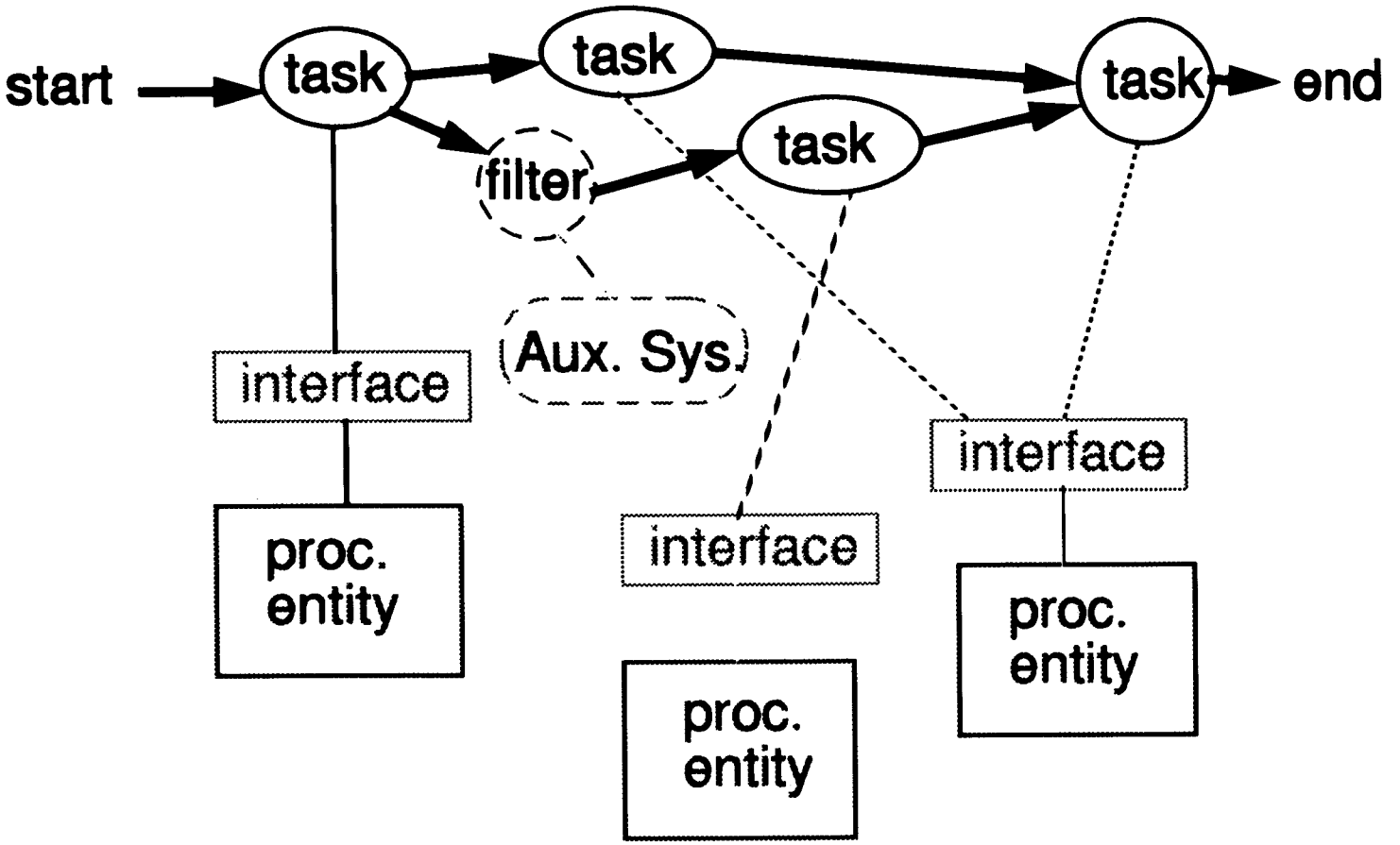
Specification of Workflows with Heterogeneous Tasks in METEOR

N. Krishnakumar
Bellcore

Amit Sheth
University of Georgia

amit@cs.uga.edu

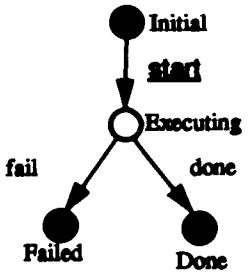
The METEOR Model for (Transactional) Workflows



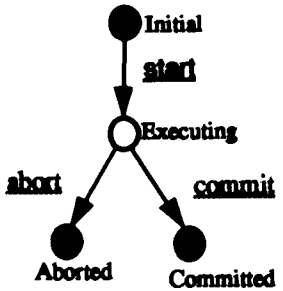
METEOR: Managing End-To-End OpeRations

Heterogeneous Tasks with different execution structures

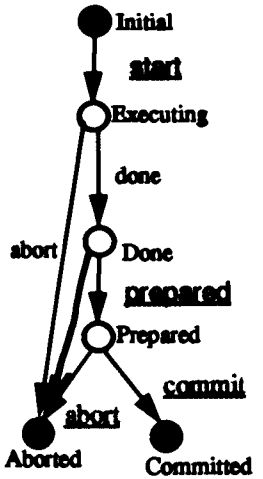
- **user tasks** involving humans in processing tasks
- **application tasks**:
  - scripts involving terminal emulations to remote systems
  - application programs/systems providing data manipulation (filters)
  - contracts (predefined interfaces) to large application systems
  - client programs or servers invoking other servers
  - database transactions



A non-transactional task



A transactional task



An open 2PC transactional task

Processing Entities

- humans (may appear as a GUI; may use document/image processing systems and applications)
- script interpreters and compilers (for processing scripts and application programs)
- (legacy) application systems (Operation Support Systems)
- servers in client-server and transaction processing systems
- DBMSs

Types of Interfaces

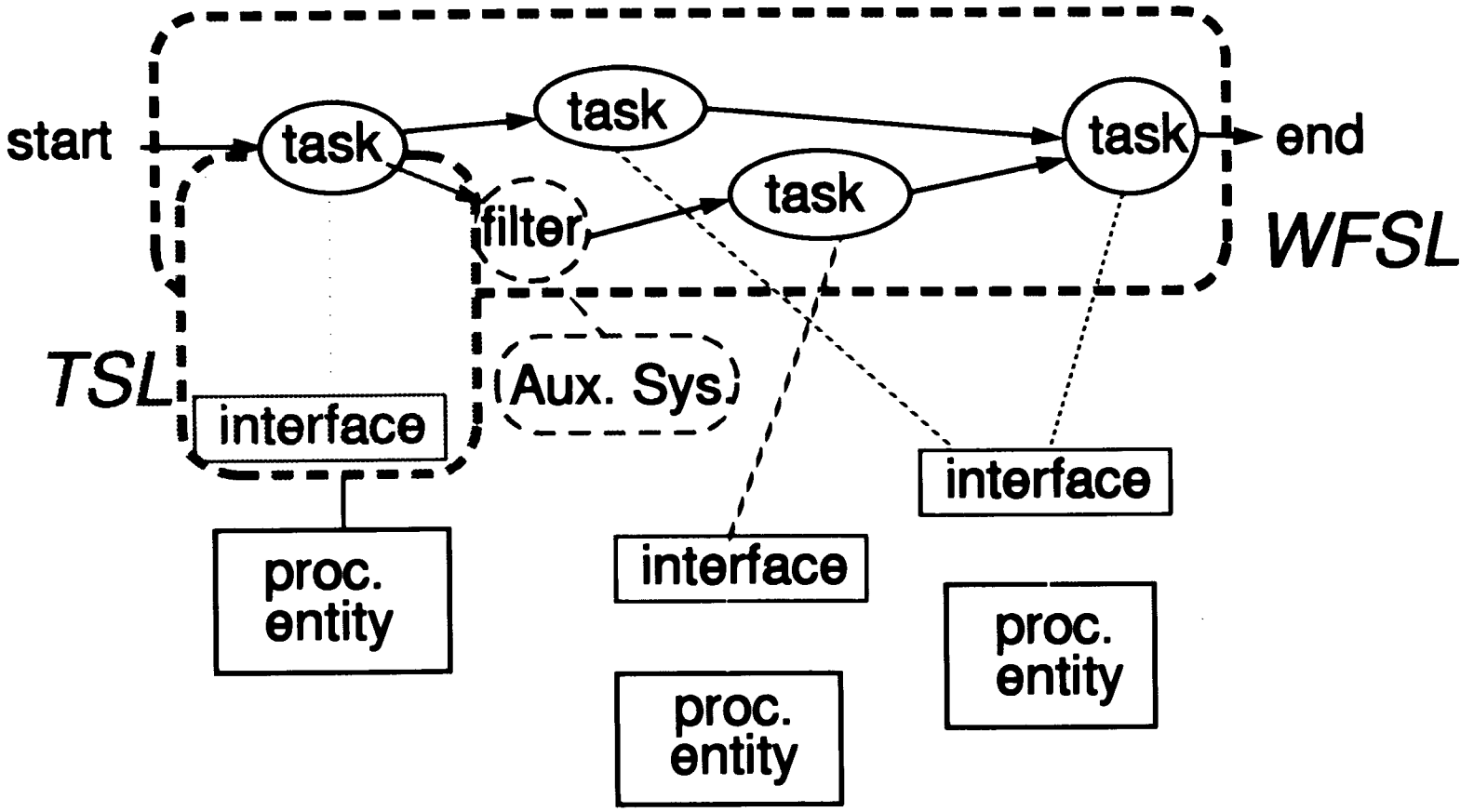
- RPC and TRPC mechanisms using transaction processing systems
- queue managers
- proprietary workstation to mainframe interfaces for
  - contracts (stored procedure calls)
  - terminal emulation

Additional Issues for Workflow Management

- 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)
- Error Handling
  - System Errors
  - Logical Errors
- Dynamic Aspects
  - processing entity not known at design time
  - new tasks & dependencies can be added dynamically
  - multiple concurrent invocation of the same task types
  - ...

An Approach to Specification :

METEOR (Sub-)Languages



WFSL: WorkFlow Specification Language
TSL: Task Specification Language

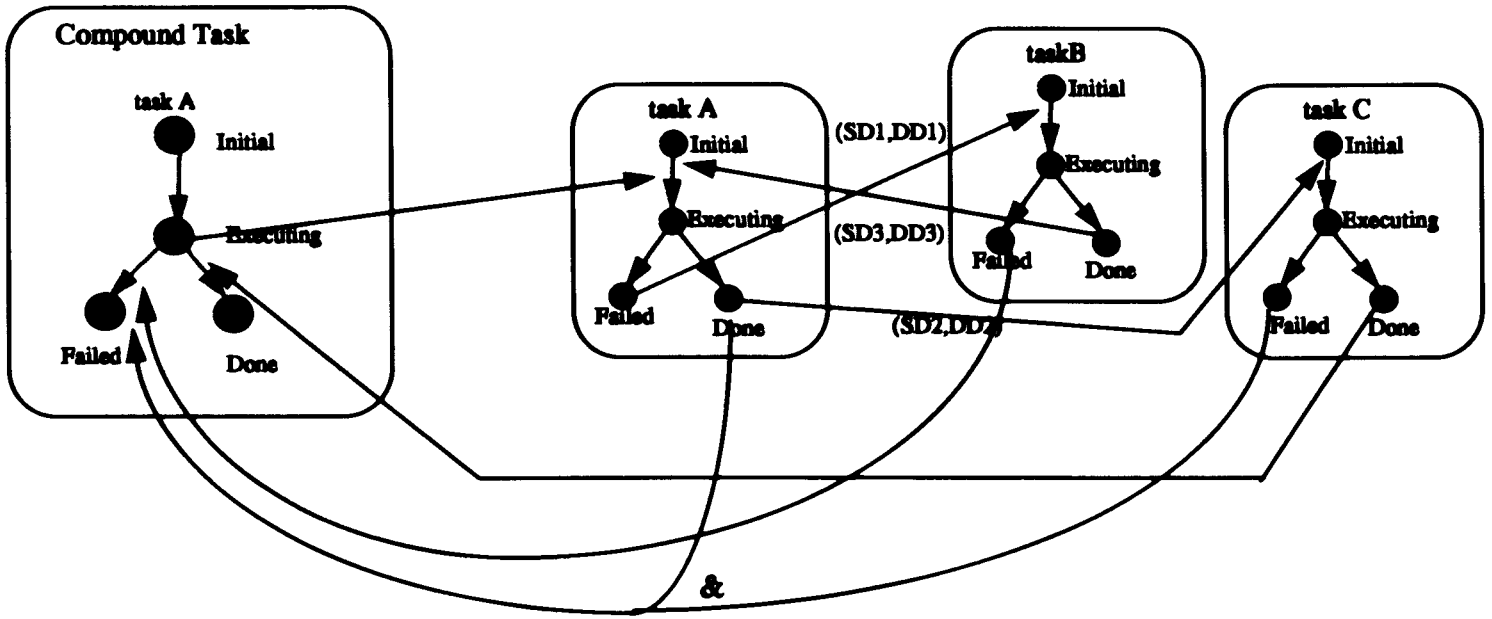
Components of WFSL (partial)

- Task types: task structures, data input/output
- Task classes, Task instances
- Declarative specification of inter-task dependencies (logical error handling)
- Data exchange statements
- Filters (interface def.)

Components of TSL(partial)

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

Workflow Example (control dependencies)



Conclusions

- We have used a generic transaction workflow model to support heterogeneous tasks and have developed high-level languages based on real applications.
  - WFSL Specification is based on a formal model
    - tasks: task structures, types, classes, instances
    - intertask dependencies, controllable transitions, ...
    - interfaces, processing entities
    - declarative control flow and data flow
    - nesting of workflows
    - handling of logical errors
  - TSL Specification provides macros conveying status of task to workflow controller, and handles system errors
- Completed two prototypes and demonstrated with a real application. Technology Transfer leading to real systems.