



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (0223) 323010
+44 223 323010
+44 223 359779
apm@ansa.co.uk**

ANSA Phase III

**November 93 TC Presentation - Extended
Transaction Framework**

John Warne

Abstract

This document contains the slides on the Extended Transaction Framework presented to the ANSA Phase III Technical Committee on 1st November 1993.

APM.1084.00.05

Draft

29 October 1993

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

Distribution:

Supersedes:

Superseded by:

Copyright © 1993 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.



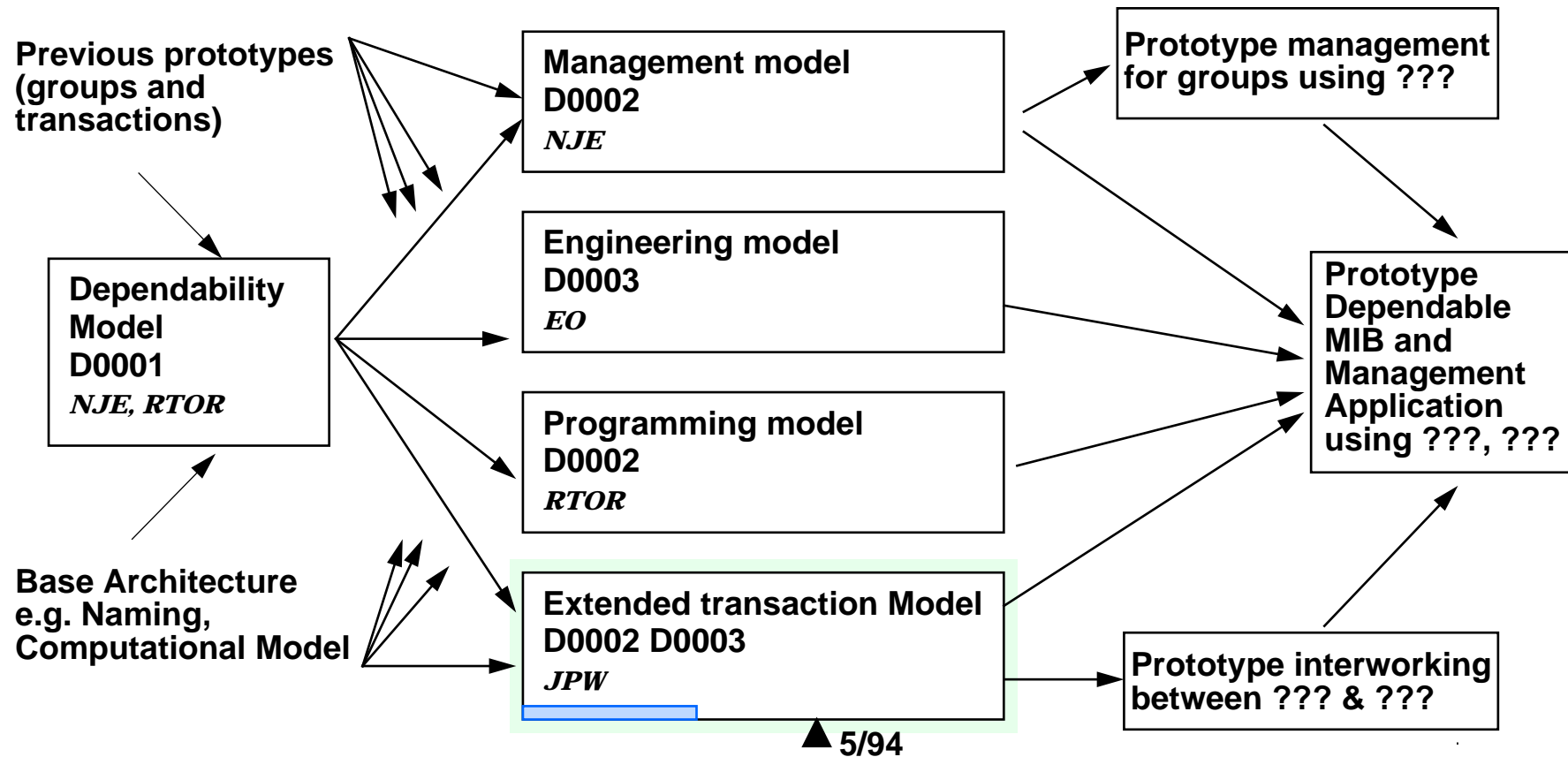
Extended Transaction Framework

Work in Progress

John Warne

Dependability Group

Dependability outline plan





Goal, Motivation, and Benefits

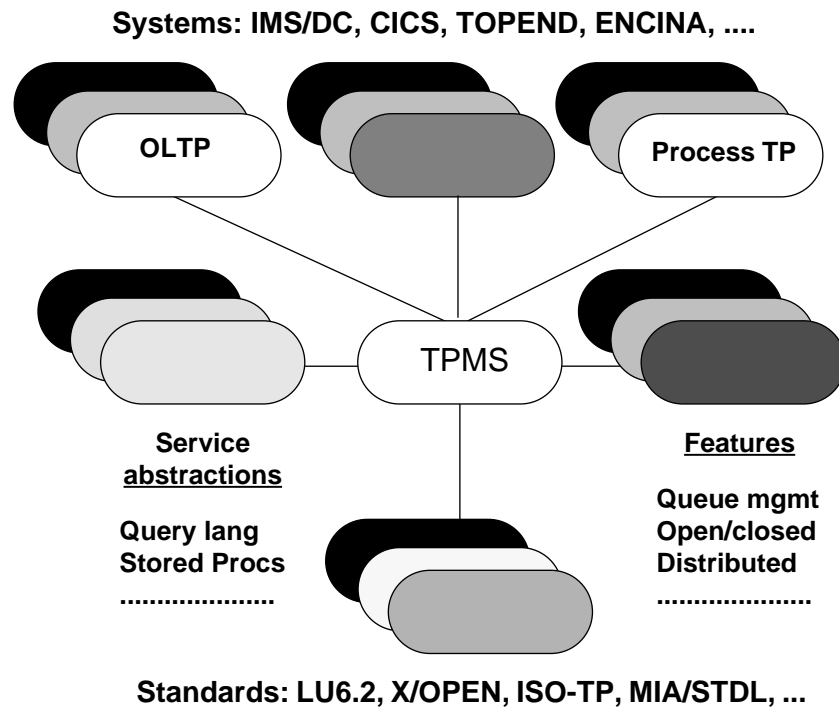
- **Goal:**
 - **Develop an architectural framework (concepts, mechanisms, and tools) for constructing and integrating multi-transaction systems in ODP environments**
- **Motivation:**
 - **Need to support a wide range of sponsor requirements:**
 - federated management of inherited TP/database systems
 - new application-specific TP models and techniques
 - interoperability between old and new TP-based applications
 - sophisticated tool kit to ease business/system construction and integration
- **Benefits:**
 - **Effective basis for building large-scale, fault-tolerant, data-intensive systems**
 - **Smooth migration strategy for evolving towards open TP services**
- **This talk presents a strategy for developing the framework**



Agenda

- **Part 1: Transaction Processing Today**
 - **Observations**
 - **Special needs**
 - **Emerging trends**
 - **Future TP characteristics**
 - **Discussion**
- **Coffee Break**
- **Part 2: Strategy for Extended Transaction Framework**
 - **Phase III approach**
 - **ACTA meta-model for reasoning about transaction models**
 - **Work flow Model for describing and constructing TP-based business processes**
 - **Rule-based triggers for integrating complex multi-transaction systems**
 - **Vision**

Observations: Heterogeneity



- Transaction Processing Manager/Monitor characteristics:

- glue to support TP-based applications
- Transaction managers vs. transaction monitors
- varying degrees of support for multi-vendor software
- Varying support for multidatabase RDA/SQL interaction with basic 1PC or cooperative 2PC
- interoperability between systems generally difficult, and sometimes impossible
- non-standard queue management
- varying degrees of availability

- Observation on standards

- increasing need for interoperability
- increasing user pressure to adopt standards
- standards evolving slowly/incomplete
- vendors developing faster than standards
- ongoing heterogeneity



Special Needs and Problems

- **Need for Interoperability**
 - **Heterogeneous transaction services**
 - **Multidatabase applications, including telecommunications network management**
- **Application-specific, semantic-based, mechanisms and policies**
 - **Concurrency**
 - **Recovery**
 - **Replication**
 - **Scheduling**
- **Problems**
 - **Some systems may not support cooperative protocols**
 - **Local autonomy of systems will typically need to be maintained**
 - **Systems may not externally communicate conflict information**
 - **Serializability may not always be possible**



Emerging Trends

- **Tool kit approach for integrating/federating multi-transaction services:**
 - **supports application-specific managers, each for a different kind of application**
 - **allows interworking between transactions with different concurrency/recovery methods**
 - **fundamental integration rule based on a two stage control sphere:**
 - **each parent tree establishes specific concurrency/recovery rules for its children**
 - **each child (as a parent) can establish another set of rules for its children**
 - **allows orderly control of resource sharing among transactions within the tree**
 - **facilitates composition of heterogeneous nested transaction trees**
- **New transaction models for complex, continuous, business processes:**
 - **long running activities**
 - **cooperative, multi-user activities**



Characteristics of Transaction Models

Model	Transaction Structure	Special subtran types	Correctness	Comments
Traditional	Flat	---	ACID	competitive
Nested	Subtrans hierarchy	Contingency Non-vital	ACID	modular
Sagas	Subtransactions	Compensating	ACD	long running
Cooperative	Subtrans hierarchy	Compensating	User-defined	CAD applications
Open Nested	Subtrans hierarchy	Compensating, Contingency, Non-vital	ACD	Publication environment
Contract flow model	Subtransactions	Compensating, Contingency, Non-vital	ACD	Long-lived, cooperating transactions
Polytrans	Hierarchy	contingency compensation	User-defined	Supports Inter-dependent data



Next Generation TP Characteristics (1)

- **Ability to relate and structure sets of ACID units to express different degrees of VCPR:**
 - **Visibility**
The degree with which one transaction can see the results of another transaction while it executing
 - **Consistency**
The correctness of the system state that a committed transaction produces
 - **Permanence**
The correctness of the system state that a committed transaction produces
 - **Recoverability**
The ability of a transaction, in the event of failure, to recover some correct system state



Next Generation TP Characteristics (2)

- **Support for existing and future transaction models**
- - **Future TP applications = flat legacy TP applications + new TP structures**
 -
 - **ACID as the fundamental law of “TP physics”**
 -
 - **Flat ACID units as the fundamental building blocks**
 -



Next Generation TP Characteristics (3)

- **Support for exploiting concurrency/recovery semantics in object systems**
- **Ability to tailor arbitrary TP structures to suit application requirements**
- **Declarative work flow language and associated tools to**
 - **describe flows of TP business processes**
 - **express dependencies between different flows**
 - **monitor, track, and query processes and dependencies**
 - **trigger transactions on detection of dependencies**
 - **dynamically manage complex TP structures and shared resources**
 - **transform declarative flows to run-time support services/platforms**



Discussion



Phase III Approach to TP Systems Integration

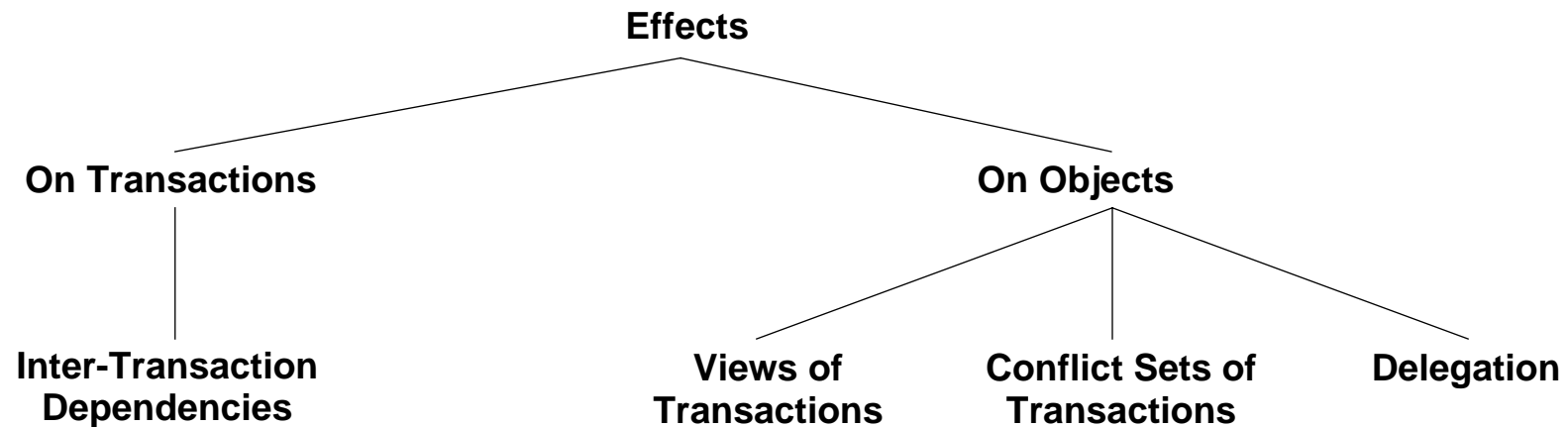
- **Not another TP model, but a toolkit for constructing TP integration services**
- **Build on the concepts and principles of existing models and abstractions:**
 - **The formal ACTA framework (reasoning about and structuring TP models)**
 - **Work flow model (specifying TP business application flows declaratively)**
 - **Active object model with rule based triggers (mechanizing TP structures)**
- **Integrate resultant TP abstractions and support mechanisms in the context of the ANSA Object Engineering, Computational and Naming Models**
- **Design transformer technology and tools**
- **Build selected prototypes to demonstrate proof of concept and workability**



ACTA Framework (1)

- **A formalism based on a set of abstract building blocks**
- **Allows the nature of complex transaction models to be studied:**
 - **assists analysis of existing transaction models**
 - **permits the construction of new transaction models**
 - **enables different transaction models to be synthesized**
 - **facilitates verification of behaviour and VCPR of complex transaction models**
- **Has been effectively used to study and construct formal specifications of several models:**
 - **flat ACID transactions**
 - **nested transactions**
 - **nested sagas**
 - **epsilon serializable transactions**
- **ACTA is a meta-model, not an architecture - identifies useful TP primitives**

ACTA Framework (2)



- Means to define significant events w.r.t. transactions and objects (e.g. begin/commit/abort)
- Open -ended framework for expressing dependencies involving significant events
- Means of controlling object visibility among transactions via views, conflicts & delegation



ACTA Framework (3)

- The model suggests two new transaction management primitives:

SetDependency (DependencyType, TargetTransactionName)

	OP1	OP2	OP3	OP4
OP1	OK	CD	NOK	NOK
OP2	NOK	AD	TD	NOK
OP3	NOK	NOK	NOK	BOCD
OP4	OK	NOK	NOK	NOK

Compatibility Table Entries:

OK - operations compatible

NOK - invoker must wait

TD - Termination dependency

AD - Abort dependency

CD - Commit dependency

BOCD - Begin-on-commit dependency

Open ended dependencies

Delegate (TargetTransactionName, ObjectSet)

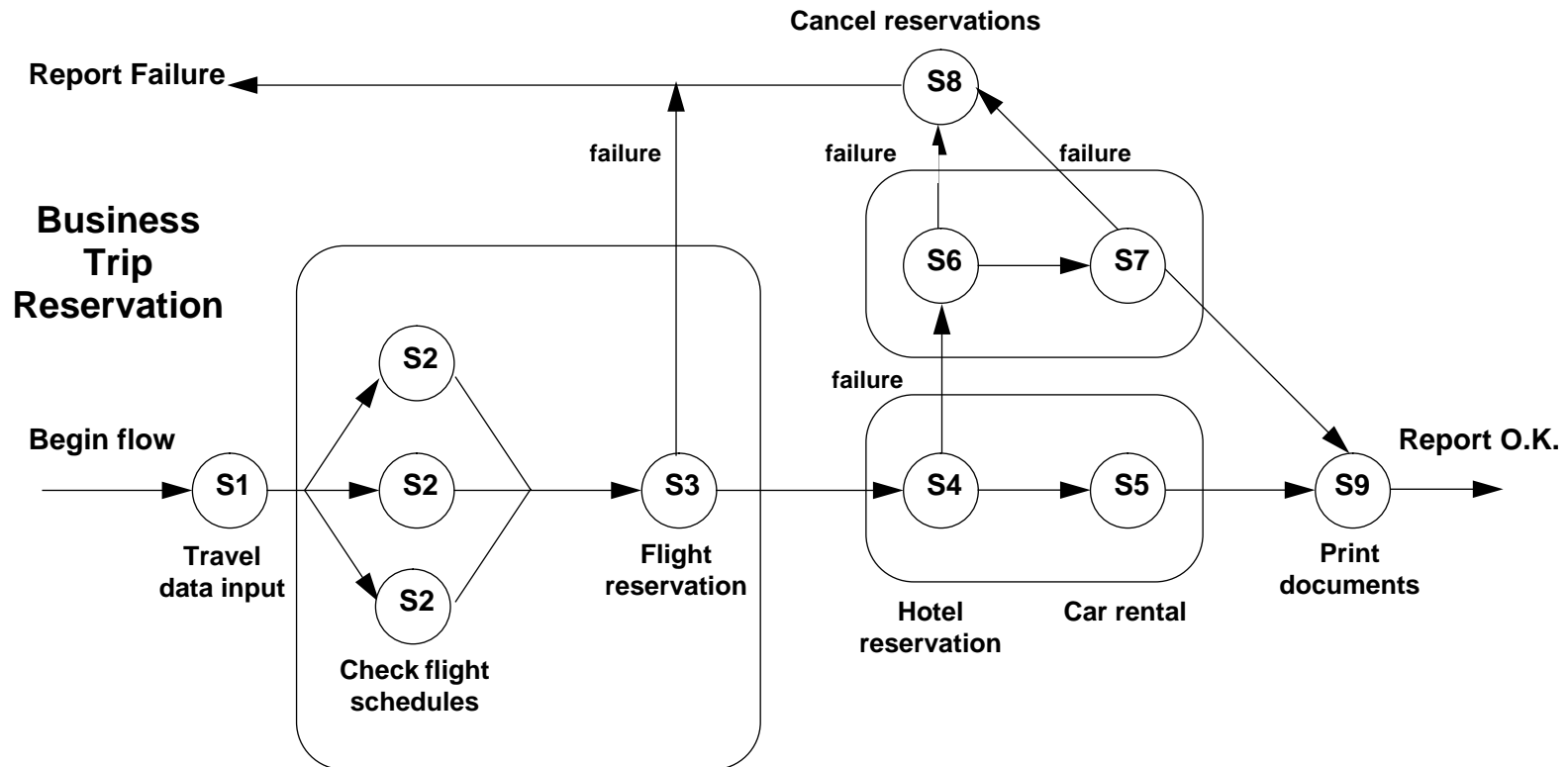
Delegates right of access of specified object set to another transaction



Work Flow Model (1)

- **Idea originates from OS JCLs**
- **Job specifications are expressed by interdependent steps**
- **Related terms in the database literature include:**
 - **multi-transaction activities**
 - **multi-system applications**
 - **application multi-activities**
 - **networked applications**
- **Work flow manager**
 - **allows dependencies to be expressed between steps**
 - **dependency evaluations are triggered by significant events**
 - **ACTA formalism can be applied to evaluate step conformities**
 - **dependencies can span across workflows**
 - **facilitates multi-transaction business process integration/federation**

Work Flow Model (2)



Active Objects and Triggers

- Events

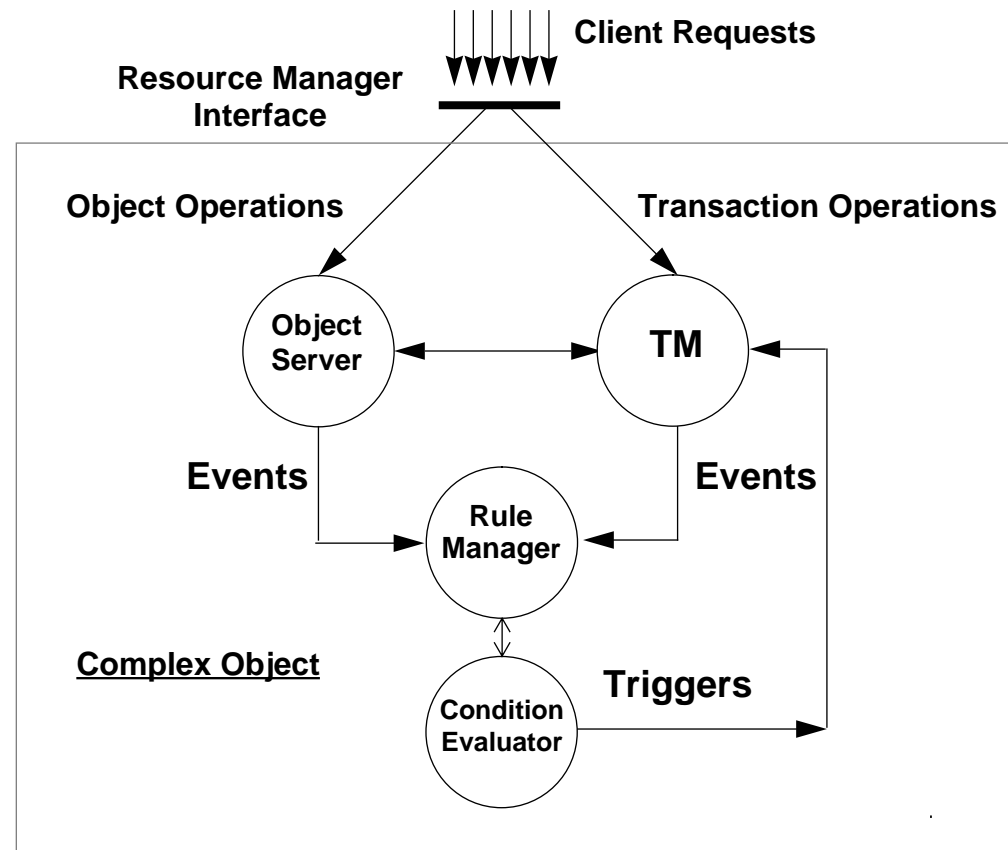
- (1) object server operations
- (2) time (absolute, relative, periodic)
- (3) transaction operations

- Rule attributes (E-C-A):

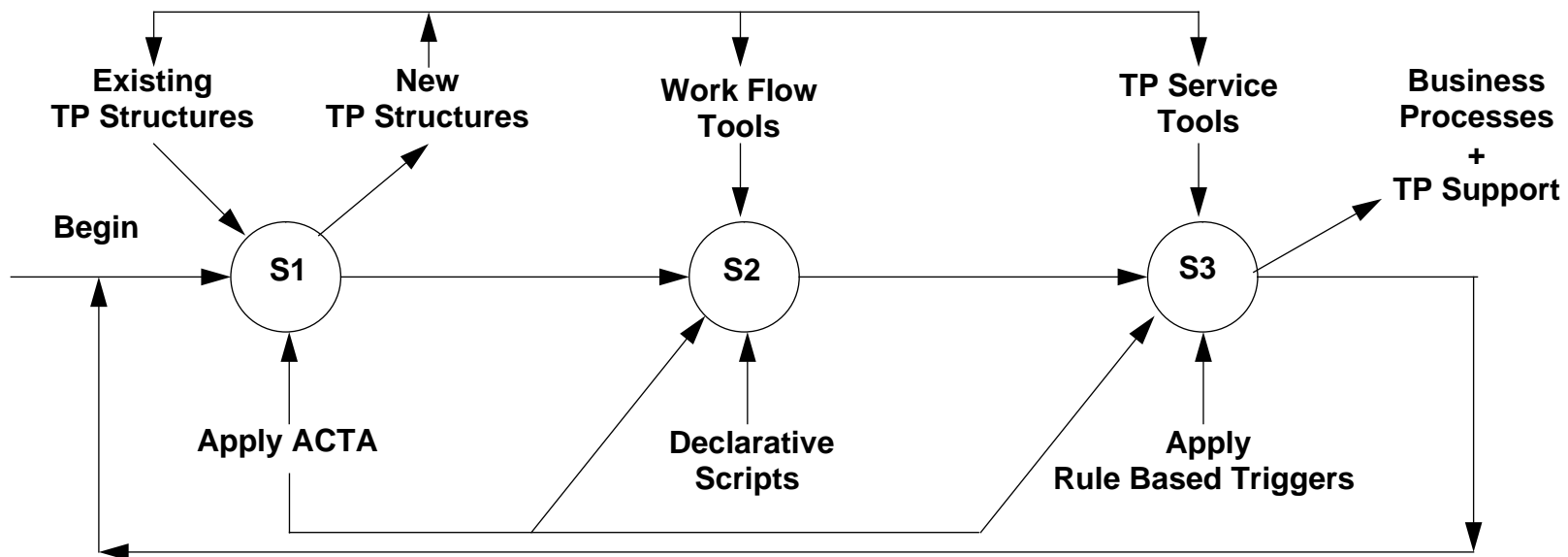
- (1) event (occurrence of interest)
- (2) condition (evaluation rules)
- (3) action (action triggered)

- Couplings:

- (1) same transaction
- (2) new transaction
- (3) immediate
- (4) deferred



Outline of vision



- S1: Construct new TP models applying ACTA concepts, formalisms and support tools
- S2: Construct business application processes using work flow scripts and support tools
- S3: Apply tools to transform declarative work flows to imperative forms



Milestones

- **Technical Overview of Extended Transaction Framework**
 - **Report available for Deptg Review Team - December 1993**
 - **Reviewed report available to the Technical Committee - February 1994**
 - **Contributes towards deliverables D0002 and D0003 - May 1994**