Distributed Transaction Processing

- introduction
- Commitment, Concurrency and Recovery (CCR)
- OSI Distributed Transaction Processing (DTP) framework
- X/Open TP model
Commitment, Concurrency and Recovery (CCR)

- background
- CCR service
- CCR protocol
- new developments

OSI DTP application layer structure
CCR standards

  - Service definition for the commitment, concurrency and recovery service element

- ISO/IEC 9805:1994
  - Protocol for the commitment, concurrency and recovery service element
    - part 1: Protocol specification (*ITU-T X.852*)
    - part 2: Protocol Implementation Conformance Statement (PICS) proforma (*ITU-T X.853*)

CCR basics

- **commitment**
  - 2-phase commit (current version)
  - presumed rollback paradigm (mimises logging requirements)

- **concurrency**
  - e.g., by a locking mechanism (or another mechanism)

- **recovery**
  - from communication failures (modelled by A-P-ABORT)
  - from end system failures (modelled by A-ABORT)
    - system crashes
    - application crashes
CCR definitions

- **atomic action**
  a set of operations characterised by the following properties:
  - atomicity (either all operations are applied or nothing is done);
  - consistency (of the bound data);
  - isolation (from operations within concurrent atomic actions);
  - durability (committed changes to bound data survive failures).

- **atomic action data**
  state and control information about an atomic action

- **bound data**
  data accessed and manipulated as part of an atomic action

**bound data**

state of distributed resource

(initial state) ➔ rollback ➔ final state

(commit)
atomic action tree

CCR service

<table>
<thead>
<tr>
<th>service</th>
<th>type</th>
<th>requested by</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-BEGIN</td>
<td>optionally confirmed</td>
<td>superior</td>
</tr>
<tr>
<td>C-PREPARE</td>
<td>non-confirmed only user data parameter</td>
<td>superior</td>
</tr>
<tr>
<td>C-READY</td>
<td>non-confirmed only user data parameter</td>
<td>subordinate</td>
</tr>
<tr>
<td>C-COMMIT</td>
<td>confirmed only user data parameter</td>
<td>superior</td>
</tr>
<tr>
<td>C-ROLLBACK</td>
<td>confirmed only userdata parameter</td>
<td>superior or subordinate</td>
</tr>
<tr>
<td>C-RECOVER</td>
<td>confirmed optionally confirmed</td>
<td>superior or subordinate</td>
</tr>
</tbody>
</table>
atomic action branch

begin atomic action branch
C-BEGIN service

- establishes an **atomic action branch**
  (between two CCR users)

- optionally confirmed

- C-BEGIN parameters:
  - atomic action identifier
    unambiguously identifies the atomic action to which the atomic action branch belongs
  - branch identifier
    unambiguously identifies the atomic action branch within the scope of the atomic action identifier
  - user data

---

prepare (optional)

Diagram showing:
- superior
- subordinate
- C-PREPARE request
- C-PREPARE indication
C-PREPARE service

- optionally used by **superior**
  (if subordinate has *not* offered commitment)

- may be substituted by application semantics

- no parameters (except user-data)

---

**offer commitment**

```
superior  subordinate
```

C-READY indication

C-READY request
C-READY service

- invoked by **subordinate** to offer commitment
- subordinate takes recovery responsibility for atomic action branch
- no parameters (except user-data)

order commitment

<table>
<thead>
<tr>
<th>superior</th>
<th>subordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-COMMIT request</td>
<td>C-COMMIT indication</td>
</tr>
<tr>
<td>C-COMMIT confirm</td>
<td>C-COMMIT response</td>
</tr>
</tbody>
</table>
C-COMMIT service

- invoked by **superior** to order commitment
- superior takes recovery responsibility for atomic action branch
- no parameters (except user-data)

**commit scenario**

```
C-BEGIN request → C-BEGIN confirm (optional)
C-PREPARE request → C-PREPARE indication (optional)
C-READY indication → C-READY request (conditional)
C-COMMIT request → C-COMMIT indication
record atomic action data
C-COMMIT confirm → C-COMMIT response
```
**rollback**

- C-ROLLBACK request
- C-ROLLBACK confirm
- C-ROLLBACK indication
- C-ROLLBACK response

---

**C-ROLLBACK service**

- used to rollback (forced “completion” of the atomic action branch)
- invoked by
  - superior to order subordinate to rollback
  - subordinate to inform superior that it refuses to proceed
- no parameters (except User-data)
rollback scenario

C-BEGIN request → C-BEGIN indication → C-BEGIN response
C-BEGIN confirm

C-PREPARE request → C-PREPARE indication → C-READY request
C-READY indication → C-ROLLBACK request
C-ROLLBACK request → C-ROLLBACK indication → C-ROLLBACK response
C-ROLLBACK confirm

recovery by superior

superior → subordinate
C-RECOVER request → C-RECOVER indication
C-RECOVER confirm → C-RECOVER response
### recovery by subordinate

- **C-RECOVER indication**
- **C-RECOVER response**
- **C-RECOVER request**
- **C-RECOVER confirm**

![Diagram showing the recovery by subordinate process]

#### C-RECOVER service

- Invoked by either superior or subordinate (after a failure)
- Optionally confirmed
- C-RECOVER parameters:
  - **recovery state**
    - ready/commit/done/unknown/retry-later
  - **atomic action identifier**
  - **branch identifier**
  - **user data**
**CCR protocol model**

```
CCR service user  
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSAP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCR APDUs</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>CCR Protocol Machine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCR service provider</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PSAP</td>
<td></td>
</tr>
</tbody>
</table>

presentation connection

PSAP Presentation Service Access Point
```

**CCR APDUs**

- **abstract syntax:**
  - ASN.1 notation

- **transfer syntax:**
  - support of ASN.1 Basic Encoding Rules (ASN.1 BER) is a **minimal** requirement
CCR APDU mapping

- different mappings
  - CCR version 1:
    » C-COMMIT APDUs \(\leftrightarrow\) P-SYNC-MAJOR
    » C-ROLLBACK APDUs \(\leftrightarrow\) P-RESYNCHRONIZE (restart)
  - CCR version 2:
    » C-COMMIT APDUs \(\leftrightarrow\) P-SYNC-MINOR
    » C-ROLLBACK APDUs \(\leftrightarrow\) P-RESYNCHRONIZE (abandon)

- concatenation of some CCR APDUs is allowed
  *(described in ASN.1 like notation)*

mapping overview for CCR protocol version 1

<table>
<thead>
<tr>
<th>CCR primitive(s)</th>
<th>CCR APDU(s)</th>
<th>Presentation primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-BEGIN req/ind</td>
<td>C-BEGIN-RP</td>
<td>P-SYNC-MINOR req/ind</td>
</tr>
<tr>
<td>C-BEGIN rsp/conf</td>
<td>C-BEGIN-RC</td>
<td>P-SYNC-MINOR rep/conf</td>
</tr>
<tr>
<td>C-BEGIN req/ind (or C-ROLLBACK req)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-BEGIN-RP</td>
<td>C-BEGIN-RC</td>
<td>P-SYNC-MINOR req/ind</td>
</tr>
<tr>
<td>C-READY req/ind</td>
<td>C-READY-RI</td>
<td>P-TYPED DATA req/ind</td>
</tr>
<tr>
<td>C-ROLLBACK req/ind</td>
<td>C-ROLLBACK-RI</td>
<td>P-RESYNCE (restart) req/ind</td>
</tr>
<tr>
<td>C-ROLLBACK rsp/conf</td>
<td>C-ROLLBACK-RC</td>
<td>P-RESYNCE (restart) rep/conf</td>
</tr>
<tr>
<td>C-ROLLBACK req/ind (or C-ROLLBACK req)</td>
<td>C-ROLLBACK-RP</td>
<td>P-RESYNCE (restart) req/ind</td>
</tr>
<tr>
<td>C-COMMIT req/ind</td>
<td>C-COMMIT-RP</td>
<td>P-SYNC-MAJOR req/ind</td>
</tr>
<tr>
<td>C-COMMIT rsp/conf</td>
<td>C-COMMIT-RC</td>
<td>P-SYNC-MAJOR rep/conf</td>
</tr>
<tr>
<td>C-COMMIT req/ind (or C-COMMIT req)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-COMMIT-RP</td>
<td>C-COMMIT-RC</td>
<td>P-SYNC-MAJOR req/ind</td>
</tr>
<tr>
<td>C-RECOVER req/ind</td>
<td>C-RECOVER-RI</td>
<td>P-TYPED DATA req/ind</td>
</tr>
<tr>
<td>C-RECOVER rsp/conf</td>
<td>C-RECOVER-RC</td>
<td>P-TYPED DATA rep/conf</td>
</tr>
</tbody>
</table>
### mapping overview for CCR protocol version 2

<table>
<thead>
<tr>
<th>CCR primitive(s)</th>
<th>CCR APDU(s)</th>
<th>Presentation primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-BEGIN req/ind</td>
<td>C-BEGIN-RI</td>
<td>P-SYNC-MINOR req/ind</td>
</tr>
<tr>
<td>C-BEGIN rsp/cnf</td>
<td>C-BEGIN-RC</td>
<td>P-SYNC-MINOR rsp/cnf</td>
</tr>
<tr>
<td>C-BEGIN req/cnf</td>
<td>C-BEGIN-RI</td>
<td>P-TYPED-DATA req/ind</td>
</tr>
<tr>
<td>C-BEGIN req/ind</td>
<td>C-BEGIN-RI</td>
<td>P-TYPED-DATA req/ind</td>
</tr>
<tr>
<td>C-READY req/ind</td>
<td>C-READY-RI</td>
<td>P-RESPynch(abandon) req/ind</td>
</tr>
<tr>
<td>C-ROLLBACK req/ind</td>
<td>C-ROLLBACK-RI</td>
<td>P-RESPynch(abandon) req/ind</td>
</tr>
<tr>
<td>C-ROLLBACK req/cnf</td>
<td>C-ROLLBACK-RC</td>
<td>P-RESPynch(abandon) rsp/cnf</td>
</tr>
<tr>
<td>C-ROLLBACK req/cnf + C-BEGIN req/ind</td>
<td>C-ROLLBACK-RI + C-BEGIN-RI</td>
<td>P-RESPynch(abandon) req/ind</td>
</tr>
<tr>
<td>C-ROLLBACK req/cnf + C-BEGIN req/cnf</td>
<td>C-ROLLBACK-RC + C-BEGIN-RC</td>
<td>P-RESPynch(abandon) rsp/cnf</td>
</tr>
<tr>
<td>C-COMPIT req/ind</td>
<td>C-COMPIT-RI</td>
<td>P-SYNC-MINOR req/ind</td>
</tr>
<tr>
<td>C-COMPIT rsp/cnf</td>
<td>C-COMPIT-RC</td>
<td>P-SYNC-MINORrsp/cnf</td>
</tr>
<tr>
<td>C-COMPIT req/ind + C-BEGIN req/ind</td>
<td>C-COMPIT-RI + C-BEGIN-RI</td>
<td>P-SYNC-MINOR req/ind</td>
</tr>
<tr>
<td>C-COMPIT req/cnf + C-BEGIN req/cnf</td>
<td>C-COMPIT-RC + C-BEGIN-RC</td>
<td>P-SYNC-MINOR rsp/cnf</td>
</tr>
<tr>
<td>C-RECOVER req/ind</td>
<td>C-RECOVER-RI</td>
<td>P-TYPED-DATA req/ind</td>
</tr>
<tr>
<td>C-RECOVER req/cnf</td>
<td>C-RECOVER-RC</td>
<td>P-TYPED-DATA req/ind</td>
</tr>
</tbody>
</table>

### new developments

- multiple commitment masters
- one-phase commit
- read-only
- cancel service
- improved error reporting
- overlapped recovery
**OSI Distributed Transaction Processing (DTP)**

- background
- OSI TP model
- OSI TP service
- OSI TP protocol
- new developments

**ACID properties of distributed transactions**

- **A** for **atomicity**
- **C** for **consistency**
- **I** for **isolation**
- **D** for **durability**

(as defined by CCR standard)
The OSI transaction Processing (TP) standard defines a **framework** for the development of transaction processing related standards (**U-ASEs**).

Example of such a standard: OSI Remote Database Access (RDA)

*Data transfer is modelled in OSI TP by means of TP-DATA, which has to be provided by one or more U-ASEs.*

*OSI TP puts constraints on the presentation services used by U-ASEs*

**Example of U-ASE: OSI RDA standard**
OSI DTP application layer structure

OSI DTP and IBM LU 6.2

OSI (TP) protocol stack

SNA (LU 6.2) protocol stack

API (CPI-C)
OSI DTP standards

- ISO/IEC 10026:1992
  Distributed Transaction Processing
  - part 1: Model (ITU-T X.860)
  - part 2: Service definition (ITU-T X.861)
  - part 3: Protocol specification (ITU-T X.862)
  - part 4: Protocol Implementation Conformance Statement (PICS) proforma (ITU-T X.863)
  - part 5: Application context proforma and guidelines when using OSI TP
  - part 6: Unstructured data transfer
  - part 7: Message queueing

OSI TP definitions

- **dialogue**
- **dialogue tree**
- **transaction** = CCR atomic action
  - transaction branch = CCR atomic action branch
  - transaction tree = CCR atomic action tree
  - transaction identifier = CCR atomic action identifier
  - transaction branch identifier = CCR atomic action branch identifier

- **TPSU, TPSUI and TPSUI-title**
- **TPSP**
application associations are used by OSI DTP for different purposes:

– for the establishment of **TP-dialogues**
  (used for performing transactions)

– for the establishment of **TP-channels**
  (used for recovery etc.)
**OSI DTP dialogue tree**

![OSI DTP dialogue tree diagram]

**TP-dialogue**

- **dialogue modes**
  - polarized control
  - shared control

- **dialogue coordination levels**
  - commitment (provider-supported transactions)
  - none (application-supported transactions)
OSI TP transaction trees

- coordination level none
- coordination level commitment

- chained transactions
- unchained transactions
**OSI TP service**

- dialogue establishment and termination
- delimitation of provider-supported transactions
- coordination of transactions
- commitment and rollback (provider-supported transactions)
- heuristic decision-making and reporting (provider-supported transactions)
- error reporting

**OSI TP functional units**

- dialogue FU
- shared control FU
- polarized control FU
- handshake FU
- commit FU
- chained transactions FU
- unchained transactions FU

(1) mutually exclusive FUs
(2) FU requires commit FU
**dialogue FU services**  
*(associated with a single dialogue)*

- dialogue establishment
- dialogue termination
- dialogue abort
- error reporting

---

**dialogue establishment**

- TP-BEGIN-DIALOGUE request
- TP-BEGIN-DIALOGUE confirm
- TP-BEGIN-DIALOGUE indication
- TP-BEGIN-DIALOGUE response
TP-BEGIN-DIALOGUE service

- establishes a dialogue with a new TPSUI
- optionally confirmed
- TP-BEGIN-DIALOGUE parameters:
  - addressing parameters (initiator and recipient)
  - functional units
  - Quality-of-Service (QoS)
  - application context name (defined in ACSE standard)
  - begin transaction (if unchained transactions FU selected)
  - confirmation (related to dialogue establishment service)
  - result: accepted/rejected (provider)/rejected (user)
  - diagnostic (if rejected by provider)
  - rollback (semantics = rollback service)
  - user data

dialogue termination

TP-END-DIALOGUE request — TP-END-DIALOGUE indication

TP-END-DIALOGUE confirm — TP-END-DIALOGUE response
**TP-END-DIALOGUE service**

- used to orderly terminate a dialogue
- optionally confirmed

**TP-END-DIALOGUE parameter:**

- **confirmation**
  - `true` (dialogue termination is conditional)
  - `false` (dialogue termination is unconditional)

---

**abort of dialogue by TPSUI**

```
TP-U-ABORT indication

TP-U-ABORT request
```
**TP-U-ABORT service**

- used to abort a dialogue
- may result in loss of indications and/or confirms to TPSUIs
- TP-U-ABORT parameters:
  - rollback (*true/false*)
  - user data

---

**abort of dialogue by provider**

- TP-P-ABORT indication
- TP-P-ABORT indication
**TP-P-ABORT service**

- used by TPSP to notify TPSUIs of the occurrence of a failure
- may result in loss of indications and/or confirms to TPSUIs
- TP-P-ABORT parameters:
  - diagnostic
  - user data

---

**error reporting service (1)**

TP-U-ERROR indication → TP-U-ERROR request
**error reporting service (2)**

- **TP-HANDSHAKE** request
- **TP-U-ERROR** indication
- **TP-U-ERROR** request
- **TP-HANDSHAKE** indication

---

**TP-U-ERROR service**

- used to notify partner TPSUI of failure
- used as a negative response to
  - handshake service
  - handshake with grant-control service
  - confirmed dialogue termination service
- no parameters
**shared control FU services**

- no services primitives associated with this FU

---

**polarized control FU services**

*(associated with a single dialogue)*

- grant control
- request control
The TP-GRANT-CONTROL service is used to give control of the dialogue to peer TPSUI. It has no parameters.
**request control**

- TP-REQUEST-CONTROL request
- TP-REQUEST-CONTROL indication

---

**TP-REQUEST-CONTROL service**

- used to request control of the dialogue from the peer TPSUI
- no parameters
handshake FU services
(associated with a single dialogue)

- handshake service
- handshake and grant control service
TP-HANDSHAKE service

◆ used by peer TPSUIs to synchronise their processing with one another

◆ TP-HANDSHAKE parameter:
  confirmation urgency
  – urgent
  – normal

handshake and grant control

TP-HANDSHAKE-AND-GRANT-CONTROL request
TP-HANDSHAKE-AND-GRANT-CONTROL confirm

TP-HANDSHAKE-AND-GRANT-CONTROL indication
TP-HANDSHAKE-AND-GRANT-CONTROL response
TP-HANDSHAKE-AND-GRANT-CONTROL service

- used by peer TPSUIs to synchronise their processing with one another and to transfer control at the same time

- TP-HANDSHAKE-AND-GRANT-CONTROL parameter:
  
  confirmation urgency
  - urgent
  - normal

commit/chained transactions/unchained transactions FUs services

- begin transaction
- deferred end dialogue
- deferred grant control
- prepare
- ready
- commit
- done
- coomit complete
- rollback
- rollback complete
- heuristic reporting
TP-BEGIN-TRANSACTION service

- used by superior TPSUI to include a subordinate TPSUI in the current transaction
- no parameters

TP-PREPARE service

- TP-PREPARE request is used by superior TPSUI to request a subordinate subtree to complete processing for the current transaction and to prepare for commitment
- TP-PREPARE indication is issued by the TPSP to intermediate or leaf TPSUIs in the transaction subtree to indicate that completion of processing for the current transaction is requested
- TP-PREPARE parameter: data-permitted (true/false)
TP-READY service

- TP-READY indication is issued by the TPSP to indicate to the superior TPSUI that the subordinate subtree is in the READY state

- no parameters

TP-COMMIT service

- TP-COMMIT request is issued by a TPSUI to indicate that all processing for the current transaction has been completed and that the transaction must be committed

- TP-COMMIT indication is issued by the TPSP to indicate, to all TPSUIs engaged in a transaction, that the outcome of the transaction is commitment and that bound data must be released in the final state

- no parameters
**TP-DONE service**

- TP-DONE request is issued by TPSUI to indicate that the bound data has been released

- TP-DONE parameter: **heuristic report**
  - heuristic mix
  - heuristic hazard

**TP-COMMIT-COMPLETE service**

- TP-COMMIT-COMPLETE indication is issued by the TPSP to all TPSUIs which are engaged in a transaction, to indicate that commitment has completed

- no parameters
TP-ROLLBACK service

- TP-ROLLBACK request is issued by a TPSUI to terminate the transaction and to release the bound data in the initial state

- TP-ROLLBACK indication is issued by the TPSP to indicate that the current transaction is being rolled back and to order release of bound data in the initial state

- no parameters

TP-ROLLBACK-COMPLETE service

- TP-ROLLBACK-COMPLETE indication is issued by the TPSP to all TPSUIs engaged in the transaction, to indicate that rollback has completed

- no parameters
**TP-HEURISTIC-REPORT service**

- TP-HEURISTIC-REPORT indication indicates (to the superior) an actual or possible occurrence of a heuristic inconsistency within the subordinate subtree

- TP-HEURISTIC REPORT parameter:
  - heuristic report
  - heuristic mix
  - heuristic hazard

**TP-DEFERRED-END-DIALOGUE service**

- used to end a dialogue with a subordinate TPSUI if the current transaction is committed

- no parameters
TP-DEFERRED-GRANT-CONTROL service

- used to grant control of the dialogue to subordinate TPSUI if the current transaction is committed
- no parameters

commitment scenario (chained transactions)
rollback scenario
(chained transactions)

OSI TP protocol

- model
- structure
- commit/rollback scenarios
- APDUs
TPPM and CPM

TPPM

- TP-service
- TPPM MACF
- SAO

CPM

- CAF service
- CPM MACF
- SAO

SAO structure

- SAO
- SACF (association management, bid, queuing, discarding)
- U-ASEs
- TP-ASE
- ACSE
- CCR
- SACF (concatenation and separation)

Universiteit Twente

29-8-98
commitment scenario (chained transactions)

OSI TP APDUs

◆ abstract syntax: ASN.1 notation

◆ transfer syntax:
  support of ASN.1 Basic Encoding Rules
  (ASN.1 BER) is a minimal requirement

◆ concatenation
  – OSI TP rules (pseudo ASN.1 notation)
  – CCR rules
  – Session Layer concatenations
rollback scenario (chained transactions)

embedding vs. concatenation

embedding

concatenation
OSI TP amendments

- read-only
- one-phase commit
- dynamic two-phase commit
- overlapped transactions
- subtransactions
- implied prepare
- cancel (accelerated rollback)
- recovery extensions
- diagnostic extensions

Message queuing
**X/Open DTP model APIs**

- **XAMTI** (*AP/CRM API*)
  - half-duplex mode
  - conversational style and “RPC” style
  - includes U-ASE
- **CPI-C** (*AP/CRM API*) - IBM LU6.2
  - half-duplex mode
  - conversational style
  - UDT (OSI DTP part 6)
- **TxRPC** (*AP/CRM API*)
  - full-duplex mode
  - RPC style
  - includes U-ASE
- **TX** (*AP/TM API*) for transaction demarcation control
X/Open DTP model SII

- **XA** (RM/TM SII)  
  allows TM to control bound data updates

- **XA+** (TM/CRM SII)  
  allows commit/rollback co-ordination between TM and CRM

- **XAP-TP** (CRM/DTP protocol SII)  
  provides access to DTP/CCR and presentation services

- **XTI** (DTP protocol/Transport SII)  
  provides access to transport network protocol stack