Summary

CoCoA:

- Separating organisational and transactional properties
- Free-style vs. strict cooperation
- Declarative specification style

Ongoing work:

- proving commutativity by mapping TM to Isabelle/HOL

Conditions on execution rules

Declaration:

```
data operation order
chapter_rule :
   forall c : chapter
   order addChapter(c);
         (editChapter(c,_) "edited" | spellCheck(c)) *;
         delChapter(c)
```

Usage:

```
when ed issues writingDone()
iff forall c : chapters
       | (query document on chapter_rule(c) <> "edited")
```

CoCoA: Integration of organisational and transactional aspects

Dependencies between enabled operations and workspace histories:

- **data operations** and **data exchange operations** must be enabled.
- a **communication** can have a condition expression that queries a workspace history.

CoCoA: Execution rules

```
data operation order
    forall c : chapter
    order addChapter(c); editChapter(c,_) *; delChapter(c)
```

CoCoA: History rules

```
history rules
    forall c : chapter
    noncommutative editChapter(c,_) and editChapter(c,_)
```

CoCoA: Data exchange operations

Declaration:

```
data exchange operations
  Chapter(c : string) =
  select addChapter(c), editChapter(c,_), delChapter(c)
```

Usage:

```
on revise enable
  ed : export Chapter("intro") to document,
       import Chapter("intro") from document
endon
```

CoCoA: Transactional aspects

Features introduced to support CoAct requirements on scenario specifications:

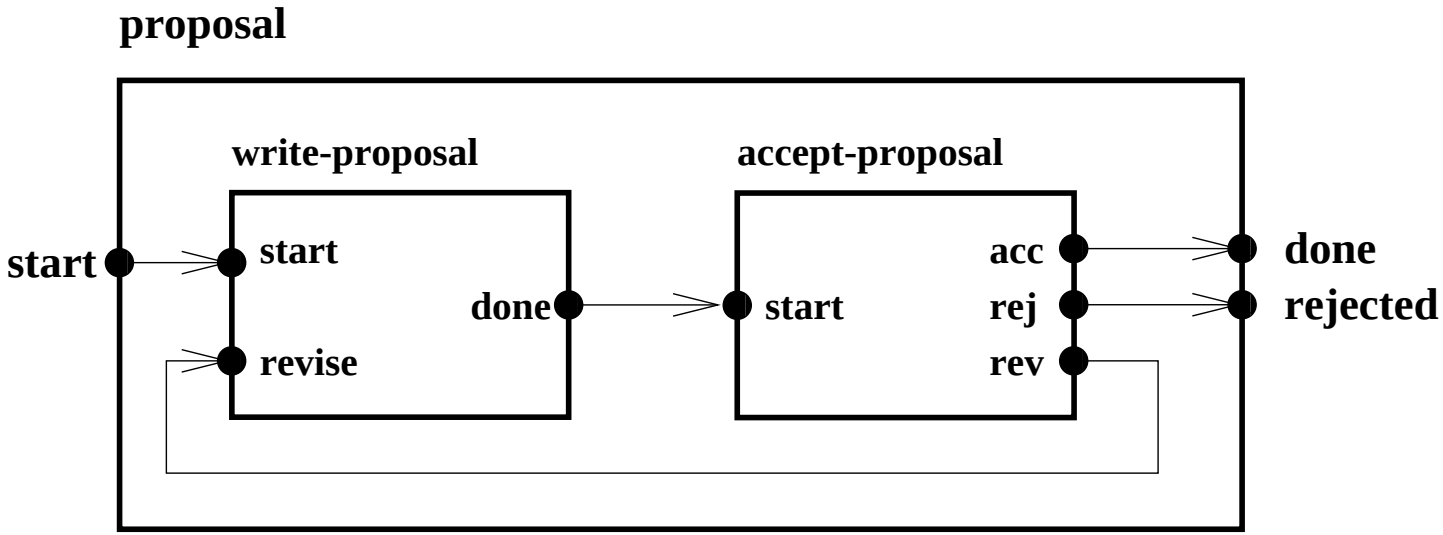
- **data exchange operations** based on workspace histories
- execution rules (keyword: **data operation order**)
- merge/commutativity rules (keyword: **history rules**)

CoCoA: Operation enabling

Declaration:

```
step write_proposal [in start, revise out done]
begin
   on start enable
      ed : addChapter("intro"),
           editChapter("intro", _),
           ...
      when ed issues introWritten() do done
   endon
end
```

CoCoA: Steps




```
step proposal [in start out done, rejected]
begin
   ...
   on write_proposal.done do accept_proposal.start
   ...
end
```

CoCoA: Organisational aspects

Steps:

- sequential, parallel, repetitive, hierarchical
- enable operations based on user roles
- many users per step, many steps per user

Communications:

- user-initiated transitions between steps
- optional parameters

Structure of a CoCoA scenario definition

```
scenario write_document
    data types ···
    data operations ···
    workspace types cda = ···
    user types ···
    communications ···
    data exchange operations ···
    procedure (ed: editor, authors : ℙ author) [in start out done]
        begin
            shared workspace document : cda
            ···
        end
    data operation order ···
    history rules ···
end write_document
```

Summary of CoCoA features

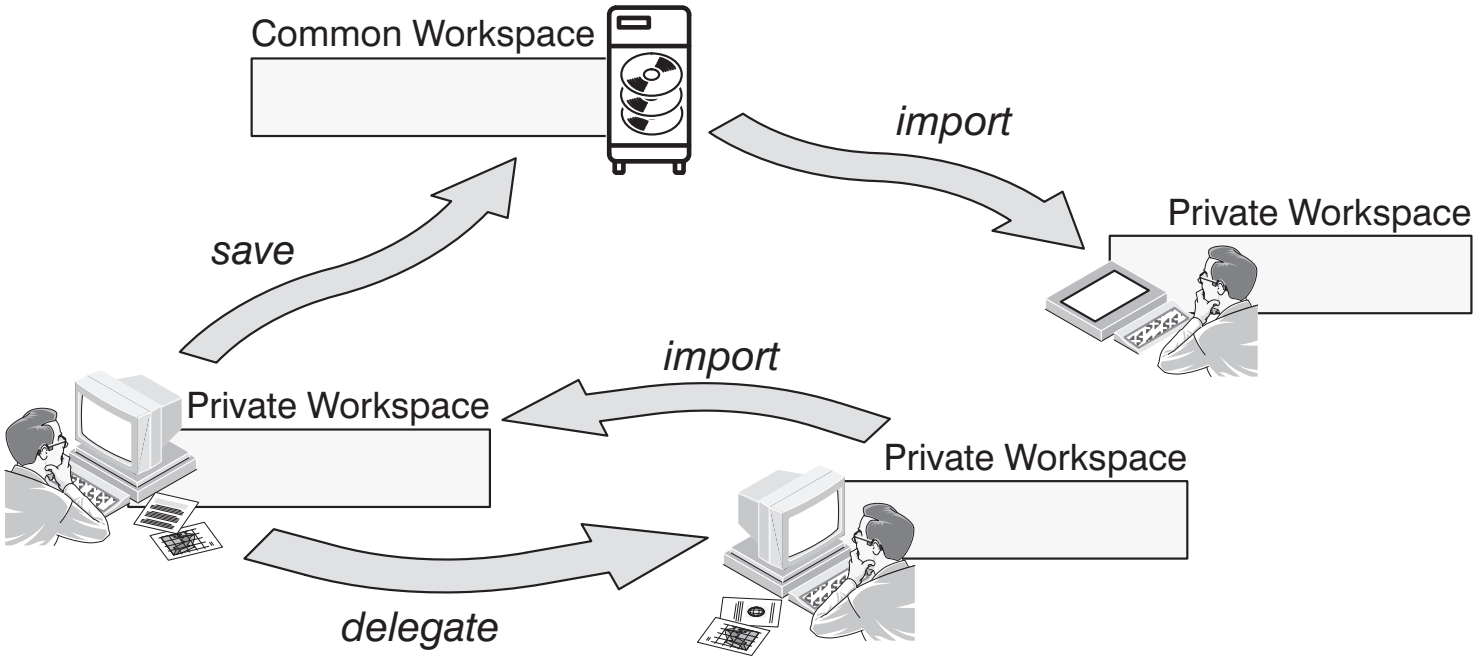
Distinction of three kinds of operation:

- data operations
- data exchange operations (import/export)
- communications

Organisational aspects: users, and work coordination

Transactional aspects: consistent data manipulation and exchange

Conceptual Organisation of the TransCoop Cooperative Transaction Model



Characteristics of CoCoA

- Designed for broad range of application domains:
  CDA, DfM, and WF
- Separation of organisational and transactional aspects
- Flexibility: not prescriptive, but descriptive
- Integrated with the CoAct transaction model
- Based on LOTOS and TM

TransCoop

ESPRIT LTR project

- GMD-IPSI, Darmstadt, Germany
- VTT, Espoo, Finland
- University of Twente, Enschede, The Netherlands

Integrating Organisational and Transactional Aspects of Cooperative Activities

Frans Faase, Susan Even, Rolf de By, Peter Apers

Sixth International Workshop on Database Programming Languages

August 20, 1997
Ester Park, Colorado, U.S.A.