Intelligent Systems

– Agent and Multiagent Technology –

Part 3

Gerhard Weiss

DKE, Maastricht University

Outline

Motivation

Agent Architectures

Coordination

- Overview
- Basic Models and Mechanisms
- Advanced Models and Mechanisms

Outline

Motivation

Agent Architectures

Coordination

- Overview
- Basic Models and Mechanisms
- Advanced Models and Mechanisms

What is Coordination?

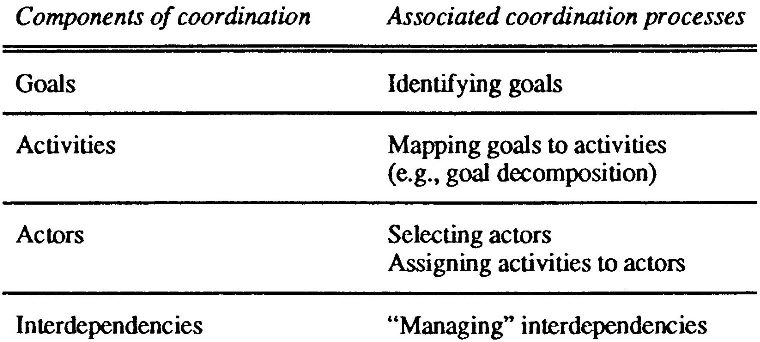
- *“Coordination is managing dependencies between activities.”* (Malone & Crowston 1994)
- *“Coordination is a special case of interaction in which agents are aware how they depend on other agents and attempt to adjust their actions appropriately.”* (Malone & Crowston 1991)
- *“Any decision by an agent that uses information concerning the existence, decisions, or decision-making strategies of other agents is a coordinated decision.”* (Stirling 1994)

Characterizing Cooperative Systems

- **Key features** in which cooperative systems differ from one another, and which are suited for characterizing such systems are:
  - **environment**: diversity, dynamics, predictability, ...
  - **cooperating entities**: number, homogeneity, goals, ...
  - **cooperation**: frequency, levels, patterns, ...
- **Space-Time Taxonomy** (Weber 1998):
  - participants are on the same vs different locations
  - participants interact at (nearly) the same vs different time

Coordination Theory (Malone & Crowston 1990)

- Components of coordination



Components of coordination	*Associated coordination processes*
Goals	Identifying goals
Activities	Mapping goals to activities (e.g., goal decomposition)
Actors	Selecting actors Assigning activities to actors
Interdependencies	“Managing” interdependencies

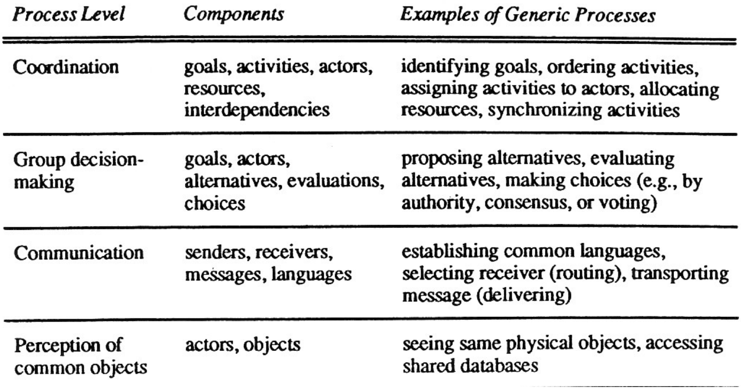
Coordination Theory (Cont'd)

- Kinds of interdependence

Kinds of interdependence	Common object	Example of interdependence in manufacturing	Examples of coordination processes for managing interdependence
Prerequisite	Output of one activity is required by the next activity	Parts must be delivered in time to be used	Ordering activities, moving information from one activity to the next
Shared resource	Resource required by multiple activities	Two parts installed with a common tool	Allocating resources
Simultaneity	Time at which more than one activity must occur	Installing two matched parts at the same time	Synchronizing activities

Coordination Theory (Cont'd)

- Processes underlying coordination



Process Level	*Components*	*Examples of Generic Processes*
Coordination	goals, activities, actors, resources, interdependencies	identifying goals, ordering activities, assigning activities to actors, allocating resources, synchronizing activities
Group decision-making	goals, actors, alternatives, evaluations, choices	proposing alternatives, evaluating alternatives, making choices (e.g., by authority, consensus, or voting)
Communication	senders, receivers, messages, languages	establishing common languages, selecting receiver (routing), transporting message (delivering)
Perception of common objects	actors, objects	seeing same physical objects, accessing shared databases

Why to Coordinate?

- **Principle of Bounded Rationality** (Simon 1957): the human mind's processing capacity is limited.
  - The amount of information that can be processed by an individual is limited
  - The detail of control an individual may wield is limited
- Increasing complexity of computer applications (distributed, open, dynamic, etc.)
- **DAI Perspective**: intelligence is not a property of isolated entities (humans, computers), but of "social" entities $\Rightarrow$ to understand intelligence requires to deal with systems being able to interact appropriately

Outline

Motivation

Agent Architectures

Coordination

- Overview
- Basic Models and Mechanisms
- Advanced Models and Mechanisms

Overview

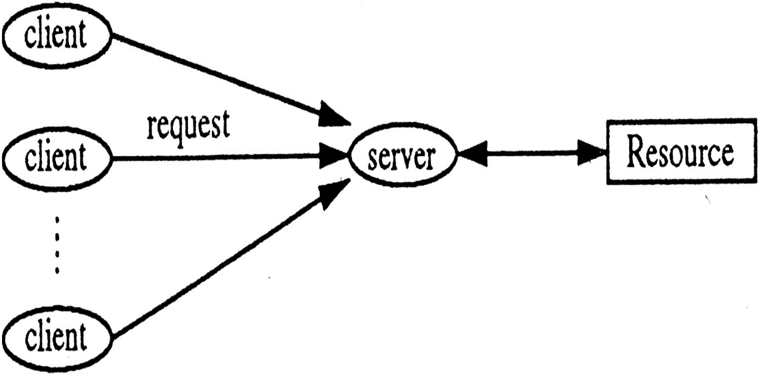
- Standard Client-Server Model
- Task and Result Sharing
- Blackboard Model
- Contract Net Model
- FA/C Principle

Standard Client-Server Model

- **Client** = process (program, application) which sends requests for operations to another process
- **Server** = process (program, application) which receives requests for operations from another process
- **Service** = the operations carried out by a server in response to request by some client
- “client” and “server” are roles which are **dynamically** played by processes
  - a process may act both as a client and as a server
  - a server may be a client of other servers (e.g., web server are often clients of local file servers managing the files in which the web pages are stored)

Standard Client-Server Model (Cont'd)

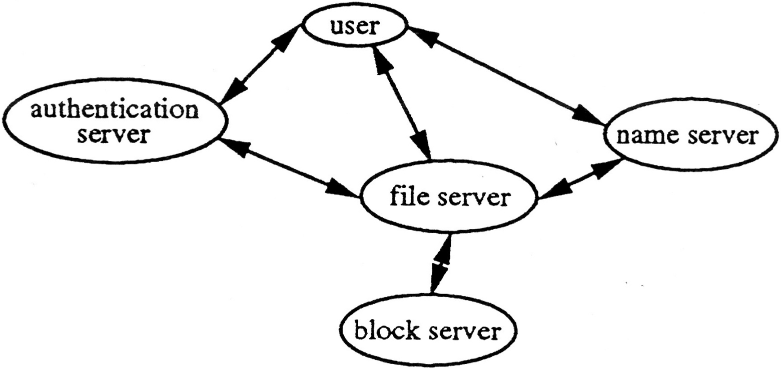
- Main idea:



(figure from (Langendörfer & Schnorr 94))

Standard Client-Server Model (Cont'd)

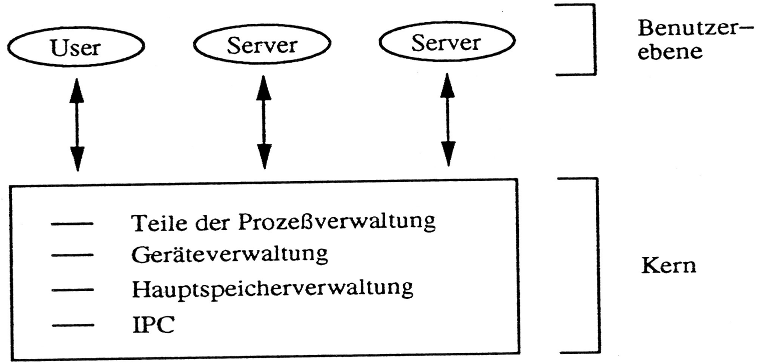
- Multiple user-client interactions:



(figure from (Langendörfer & Schnorr 94))

Standard Client-Server Model (Cont'd)

- clients and servers operate at the same level, above some (efficient) operating system kernel:



(figure from (Langendörfer & Schnorr 94))

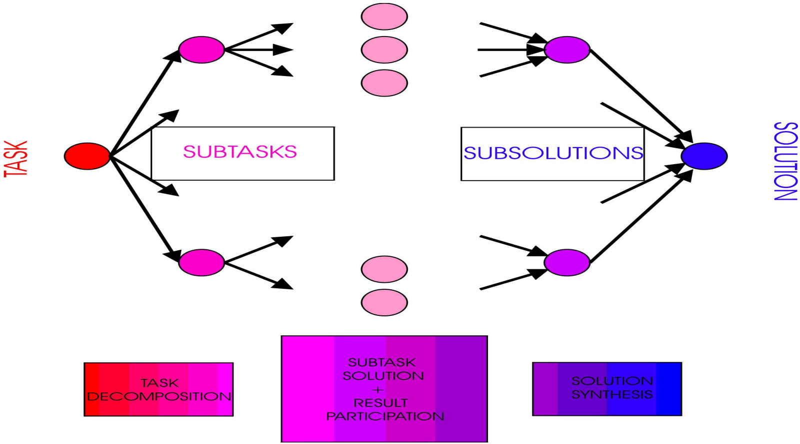
Standard Client-Server Model (Cont'd)

- Pros and Cons:
  - ⊕ simple control structure
  - ⊕ simple synchronization
  - ⊖ server may turn out as bottleneck (centralization)
  - ⊖ poor failure tolerance (centralization)
  - ⊖ information used by server(s) may be outdated (e.g. due to communication delay)

  ⇒ server replication (but: requires coordination and synchronization on its own), hierarchical control structures

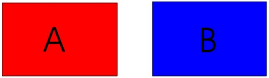
Task and Result Sharing

- Underlying **generic cooperation model**:

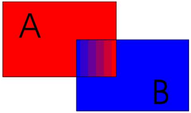


Task and Result Sharing (Cont'd)

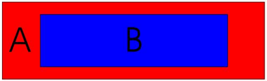
- How (sub-)tasks and (sub-)problems can be related to each other:



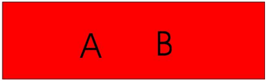
disjoint



overlapping



subsuming



identical

Task and Result Sharing (Cont'd)

- Task and result sharing offer **potential advantages**:
  - each subproblem can be solved with less knowledge
  - each subproblem requires less resources
  - parallelism and robustness
  - use of multiple sources of knowledge and skills
  - mutual support through exchange of pre-results

⇒ each of these potential advantages requires design efforts on its own!

Task and Result Sharing (Cont'd)

- **Key challenge** raised by task and result sharing: "Which agent is responsible for which part of the overall cooperation process?" (connection problem)
  - What needs an involved entity to know?
  - Efforts and costs of interaction and its control?
  - Level of task decomposition?
  - Strategies for synthesis?

  ⇒ answers depend on relationships between (sub-)tasks/problems

Blackboard Model

- Underlying idea:

- General requirements:
  - common memory
  - read/write control

Blackboard Model (Cont'd)

- **Characteristics:**
  - every participant reads from and writes on common memory area
  - participants may W/R independently or in a coordinated way
  - address of sender needs not to be known
  - participants themselves decide on information anncouncement (whether, when, ...)
  - participants themselves decide on information search and evaluation
  - suited for *open* applications
  - supports *variability in expertise*

Blackboard Model (Cont'd)

- **Characteristics (Cont'd):**
  - flexible w.r.t. representation (data) structure, but FIXED
  - regionalization/structuring is possible, if required for reasons of efficiency and effectiveness
  - suited for event- and data-driven applications
  - supports incremental generation of solutions
  - typically used as a component in knowledge-based systems
  - first realization: HEARSAY-II (1980)

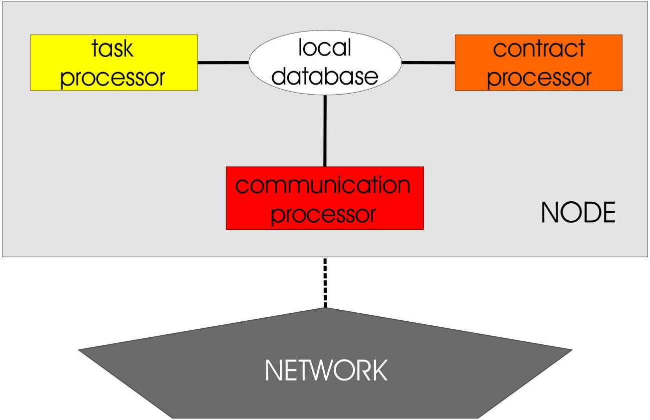
Contract Net Model

- **General characterization:**
  - **Network of nodes** (cooperating units) acting as managers and contractors
  - A **manager** announces tasks to be done
  - A **contractor** bids for right to carry out task
  - The contractor responding with the best bid is selected from the announcing manager

⇒ flexible and distributed control,
dynamic roles (agent can act as manager and contractor)

Contract Net Model (Cont'd)

- **Node architecture:**



Contract Net Model (Cont'd)

- **Negotiation steps**
  1. *Task announcement*
     - eligibility specification (minimal requirements on potential contractor)
     - task abstraction (short description)
     - bid specification (its structure and contents)
     - expiration date
  2. *Bidding*
     - response in accordance with bid specification
  3. *Contracting*
     - selection of best bid according to some criteria

Contract Net Model (Cont'd)

- **Final remarks:**
  - Key questions a designer (or contractors/managers) need to answer:
    - What tasks should be announced? (Reasons why an entity should do a task on his/her own?)
    - Who should receive a specific announcement?
    - Why should a potential contractor bid?
    - Selection criteria for managers in case of multiple bids?
    - Selection criteria for contractors in case of multiple announcements?
  - Conceptually the contract net is located between master-slave and blackboard models (predominance of manager resp. contractor)

FA/C Principle

- Functionally accurate cooperation (FA/C) as a general design guideline for cooperation when the individuals' local knowledge is incomplete, uncertain and inconsistent
- "If available information is not perfect, do not longer aim at building a system in which only completely accurate information is exchanged among cooperating entities. Instead, in response to this lack of perfection make sure that the involved parties exchange functionally correct information (tentative partial results) and that they cooperate in refining these information."
  - *functionally correct* = acceptable and reasonable from a party's local point of view
  - *cooperation* = iterative refinement, transformation of local into global correctness

FA/C Principle (Cont'd)

- FA/C requires that an involved party is able to
  - measure and evaluate functional correctness
  - detect inconsistencies (etc) between its tentative partial results and those received from others
  - integrate into its local database those portions of partial results which are consistent with its own results
  - revise and extend its tentative partial results on the basis of the newly integrated data

Outline

Motivation

Agent Architectures

Coordination

- Overview
- Basic Models and Mechanisms
- Advanced Models and Mechanisms

Overview

- Auctioning and Voting
- Negotiation
- Joint Planning
- Commitments and Conventions

Auctioning and Voting

- **English (first-price open-cry) auction:**
  - bidder is free to raise his bid (for a predefined amount)
  - auction ends when no bidder is willing to raise anymore
  - highest bidder wins at the price of his bid
  - variants: open-exit (no reentry after declaring to exit), correlated value (auctioneer increase price at constant rate)
- **first-price sealed-bid auction:**
  - each bidder submits one bid without knowing the others' bids
  - highest bidder wins and pays amount of his bid

Auctioning and Voting (Cont'd)

- **Dutch (descending) auction:**
  - seller continuously lowers price
  - auction ends when one of the bidders takes the item at the current price
- **Vickrey (second-price sealed-bid) auction:**
  - each bidder submits one bid without knowing the others
  - highest bidder wins, but at a price of the second-highest bid
- Japanese auction (*n* lowest bidders are excluded), combinatorial auctions (simultaneous bidding for different items), leveled-commitment auctions, ...

Auctioning and Voting (Cont'd)

- Generally characterizing auction settings, I: (in)dependent valuation
  - "**private value auction**": each bidder's value of the good depend only on her/his own preferences (e.g., no resell)
  - "**common-value auction**": each bidder's value entirely depends on other agents' values
  - "**correlated value auction**": bidder's value depends partly on own preferences and partly on others' values

Auctioning and Voting (Cont'd)

- Generally characterizing auction settings, II: risk attitude
  - "**risk-averse bidder**": bidder who would prefer to get the good even if she/he paid slighly more for it than her/his private value
  - "**risk-averse auctioneer**": auctioneer who prefers to sell the good even if at a lower price than he could achieve under different circumstances
  - "**risk-neutral**": neither risk-averse nor prepared to take a risk

Auctioning and Voting (Cont'd)

- **Expected revenue, I:**

  English/Dutch/Vickrey/First-price-sealed-bid produce the same expected revenue to the auctioneer in private value auctions where the values are independently distributed and bidders are risk-neural

- **Expected revenue, II:**

  Among risk averse bidders, Dutch and First-price-sealed-bid give higher expected revenue to the auctioneer than Vickrey or English

  (because bidder can 'insure' himself by bidding more than would be offered by a risk-neural bidder)

Auctioning and Voting (Cont'd)

- **Expected revenue, III:**

  Risk-averse auctioneers achieve higher expteced revenue via Vickrey or English than via Dutch or First-price-sealed-bid

- **Critical issues raised by auctions:**
  - Lying auctioneers (e.g., overestimate of 2nd-highest bid in Vickrey, use of shills to increase bidders' valuations)
  - bidder collusion
  - counterspeculation (process of obtaining information either about the true value of the good, or about the valuations of other bidders)
  - there are results in auction theory, treated in ongoing research on electronic auctions

Auctioning and Voting (Cont'd)

- **Voting based on plurality protocols:**
  - all alternatives are compared simultaneously
  - alternative with highest number of votes wins

  - **Problem:** irrelevant alternative can split majority
    - some voters stay in favor of old favorite, and some are in favor of the new alternative
    - it may even be the case that the old favorite and the new irrelevant alternative drop below one of the originally less preferred alternatives

Auctioning and Voting (Cont'd)

- **Voting based on binary protocols:**
  - alternatives are voted on pairwise
  - winner stays, and will be compared to another alternative
  - surviving alternative is final winner

  - **Problem 1:** irrelevant alternatives can change outcome
  - **Problem 2:** agenda (i.e., order of pairings) is crucial

Auctioning and Voting (Cont’d)

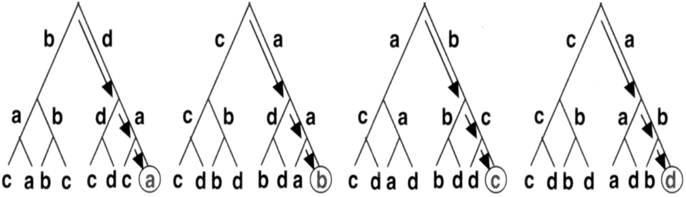
- **Voting based on binary protocols (Cont’d):**

  Illustration: influence of agenda

35% of agents have preferences $c \succ d \succ b \succ a$

33% of agents have preferences $a \succ c \succ d \succ b$

32% of agents have preferences $b \succ a \succ c \succ d$



Auctioning and Voting (Cont'd)

- **Borda protocol**
  - each voter generates his own preference list over available alternatives ("alternative 1 first, then alternative 5, then ...")
  - if there are $|A|$ alternatives, then an alternative gets $|A|$ points whenever it is highest in some preference list, $|A| - 1$ whenever it is second, and so forth
  - alternative with highest total count becomes social choice

  - **Problem:** adding and removing irrelevant alternatives

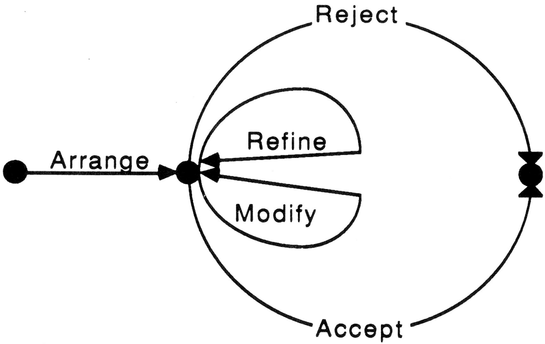
Auctioning and Voting (Cont'd)

- **Borda protocol (Cont'd)**
  - Illustration:

Agent	Preferences
1	$a \succ b \succ c \succ d$
2	$b \succ c \succ d \succ a$
3	$c \succ d \succ a \succ b$
4	$a \succ b \succ c \succ d$
5	$b \succ c \succ d \succ a$
6	$c \succ d \succ a \succ b$
7	$a \succ b \succ c \succ d$
Borda count	*c* wins with 20, *b* has 19, *a* has 18, *d* loses with 13
Borda count with *d* removed	*a* wins with 15, *b* has 14, *c* loses with 13

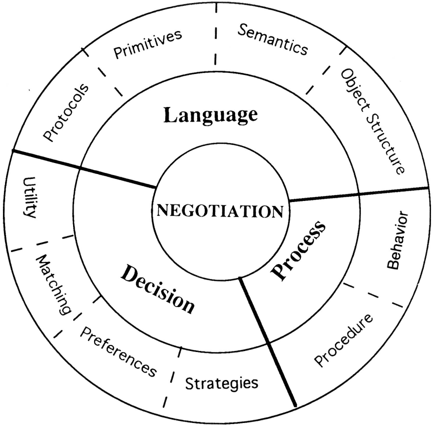
Negotiation I: Getting Acquainted

- Negotiation = exchange of information in multiple rounds for the purpose to come to an agreement
- (Computational) negotiation is simple ...



Negotiation I: Getting Acquainted (Cont'd)

- ... (Computational) negotiation is simple, isn't it? – Categories of negotiation:



Negotiation I: Getting Acquainted (Cont'd)

- **Negotiation language**: communication primitives for negotiation, semantics, structure of negotiation topics, ...
- **Negotiation decision**: estimation of utility of negotiation, strategies for negotiation, preferences over negotiation topics, ...
- **Negotiation process**: context of negotiation, analysis of negotiation effects, ... (macro perspective)

- Negotiation primitives:
  - initiators: propose, arrange, request, inform, query, ...
  - reactors: answer, refine, modify, change, bid, reply, ...
  - completers: confirm, promise, commit, accept, reject, ...

Negotiation II: A Basic Model

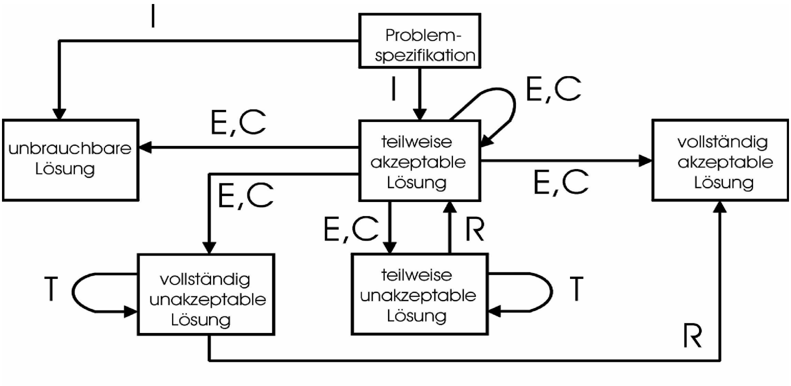
- Based on negotiated search approach (Lander 1992), aims at successively refining partial solution(s)
- **Key assumptions:**
  - conflicts result in communication
  - conflict based on violation of strong constraints: drop partial solution
  - conflict based on violation of weak constraints: partial solution = potential compromise
  - explore potential solutions in parallel
- Not covered by this basic model (among other things): multi-linked negotiation (multiple interconnected negotiation issues (Zhang/Lesser/Abdallah))

Negotiation II: A Basic Model (Cont'd)

- **Operations** available to the cooperating entities:
  - initiate-solution (start of iterated search)
  - critique-solution (evaluation)
  - extend-solution (consistent extension)
  - relax-solution-requirement (relaxation of weak constraints)
    - unilateral relaxation
    - feedback-based relaxation
    - problem-state relaxation
  - terminate-search (conclude iterated search)

Negotiation II: A Basic Model (Cont'd)

- The basic model at a glance:

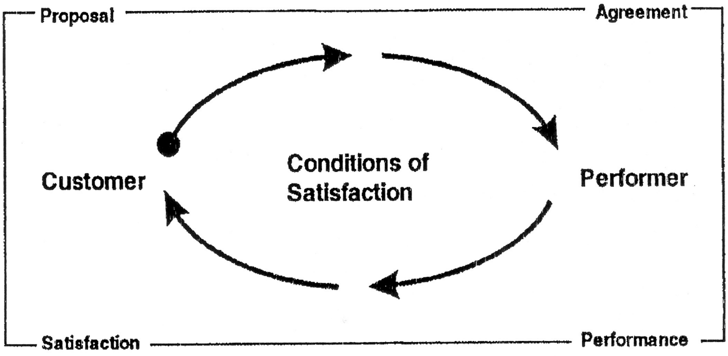


I
Problem-spezifikation
I
E,C
unbrauchbare Lösung
E,C
teilweise akzeptable Lösung
vollständig akzeptable Lösung
E,C
E,C
E,C
R
T
vollständig unakzeptable Lösung
teilweise unakzeptable Lösung
T
R

(I=initiate, C=critique, E=extend, R=relax, T=terminate)

Negotiation III: Consumer-Performer Model

- Proposed by (Medina-Mora, Winograd, Flores & Flores 1992) as an approach to workflow management
- Basic idea: the action workflow loop



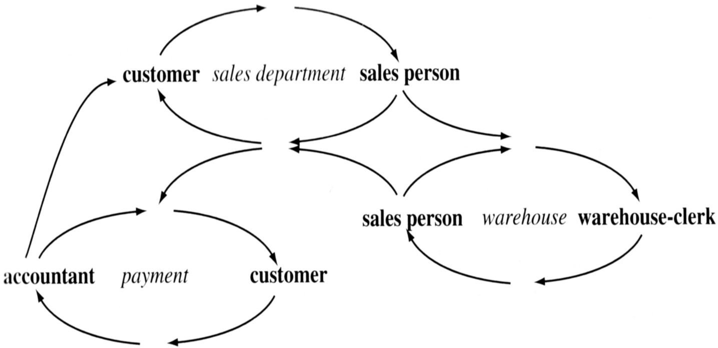
(figure from (Medina-Mora et al. 1992))

Negotiation III: Consumer-Performer Model (Cont'd)

- The loop's four phases:
  - *Proposal*: the customer requests (or performer offers) completion of a particular action
  - *Agreement*: the two parties come to mutual agreement on the conditions of satisfaction
  - *Performance*: performer declares to the customer that the action is complete
  - *Satisfaction*: the customer declares to the performer that the completion is satisfactory
- At any phase, there may be additional actions, such as clarifications, further negotiations about conditions, changes of commitments (e.g., w.r.t. to delivery time)

Negotiation III: Consumer-Performer Model (Cont'd)

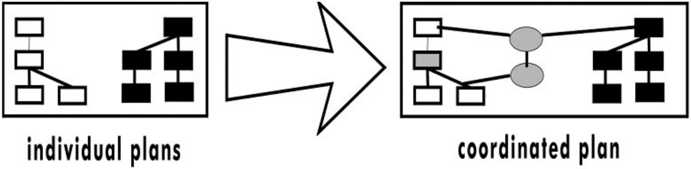
- Chain of workflow loops



(figure from (Borghoff & Schlichter 2000))

Joint Planning, I: General Issues

- **Basic idea**



Joint Planning, I: General Issues (Cont'd)

- **Key design questions:**
  - What exactly means "coordinated" plan in a given application?
  - How can parties detect "(un)coordinatedness"?
  - How to jointly generate coordinated plans?
- **General Taxonomy of planning:**
  - single-component approaches
  - multi-component approaches
    - single planner + multiple executors
    - multiple planners + single executor
    - multiple planners + multiple executors

Joint Planning, I: General Issues (Cont'd)

- **Relationships among plans**
  - *positive:*
    - equality (plans have same effects)
    - subsumption (effects of plan A cover effects of plan B)
    - favorableness (minor modification of plan A reduces efforts for carrying out plan B)
  - *negative:*
    - resource conflicts
    - incompatibility of activities and states

Joint Planning, II: PGP

- **Characteristics of Partial Global Planning PGP** (Durfee 1988)
  - general coordination schema
  - no assumptions about distribution of subproblems, expertise, or resources
  - follows the principle of functionally accurate cooperation (FA/C)
  - basic idea: each involved party can represent and reason about the actions and interactions of other involved parties and how they affect local activity (→ "Partial Global Plans")

Joint Planning, II: PGP (Cont'd)

- Partial Global Plan specifies how different *parts* of a whole *plan* to achieve more *global* states
- Components of a Partial Global Plan
  - **objective**: why PGP exists, including its goal
  - **plan-activity map**: what the parties are doing, major current plan steps (including costs and expected results)
  - **solution-construction-graph**: information about how the parties should interact, what results should be exchanged and when to exchange them
  - **status**: bookkeeping information (pointers to relevant information received from other parties)

Joint Planning, II: PGP (Cont'd)

- Key limitations of partial global planning
  - local actions may be executed without joint agreement
  - plan coordination is based on a relatively simple level of abstraction (e.g., no distinction between short- and long-term plan, fine- and coarse-grained plans)

Commitments and Conventions

- “Commitment” as a **first-class modeling abstraction**
- **Types of Commitments**
  - Psychological commitment: commitment to oneself (to one’s own intentions)
  - Social commitment: commitment to others (to do certain actions or to prevent certain conditions)
  - Joint commitment: commitment of multiple actors to a joint action
  - Precommitment: decision to get get involved in a certain commitment in the future
  - Leveled commitment: relative, rather than absolute commitment (commitment may be canceled, perhaps for the price of some penalty)

Commitments and Conventions (Cont'd)

- **Operations on commitments**
  - **Create**: instantiates a commitment
  - **Discharge**: satisfies the commitment (success case)
  - **Cancel**: revokes the commitment (failure case)
  - **Release**: elimination of the commitment (no matter of success and failure)
  - **Delegate**: shifts the role of debtor to another agent
  - **Assign**: transfers a commitment to another creditor

Commitments and Conventions (Cont'd)

Formally capturing commitments (Singh 1999)

A commitment C is a four-place relation involving a proposition (p), two individual agents (x, y) and a group of agents (G): $c = C(x, y, G, p)$ denotes a commitment from x toward y in the context of G and for the proposition p.

where

x is the debtor (commiter), y is the creditor (commitee), G is the context group, p is the discharge condition of commitment c.

Commitments and Conventions (Cont'd)

- "Convention" as a **first-class modeling abstraction**
- Convention = decription of the circumstances under which an actor should (or is allowed to) reconsider its commitments
- **Why conventions are needed:**
  - between making a commitment and the associated intention being carried out, the "world" may change (significantly) $\Rightarrow$ satisfied, unattainable, ...
  - actor must be able to react on changes
- **Challenges:**
  - decrease of reliability, if commitments are rectified or abandoned too often
  - balance between constantly and never reconsidering commitments

Commitments and Conventions (Cont'd)

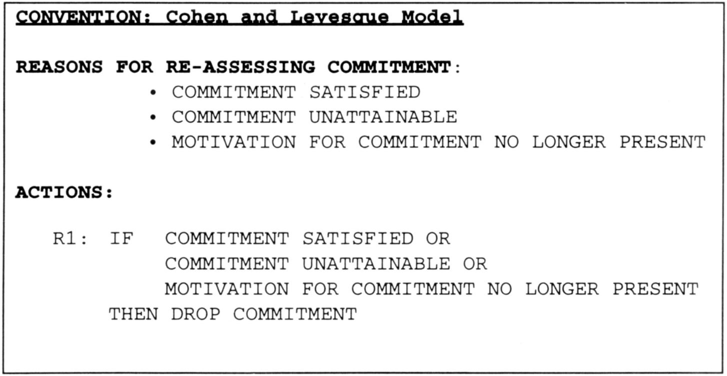
- **Social convention** = description of how to behave with respect to other community members when commitments alter. (For example, to inform them, to offer alternatives, etc.)
- "**Centrality of Commitments & Conventions Hypothesis**" (Jennings 1993):

  > *All coordination mechanisms can ultimately be reduced to (joint) commitments an their associated (social) conventions.*

  ⇒ "coordination = commitments + conventions +
  social conventions + local reasoning"

Commitments and Conventions (Cont’d)

- **Example 1:**

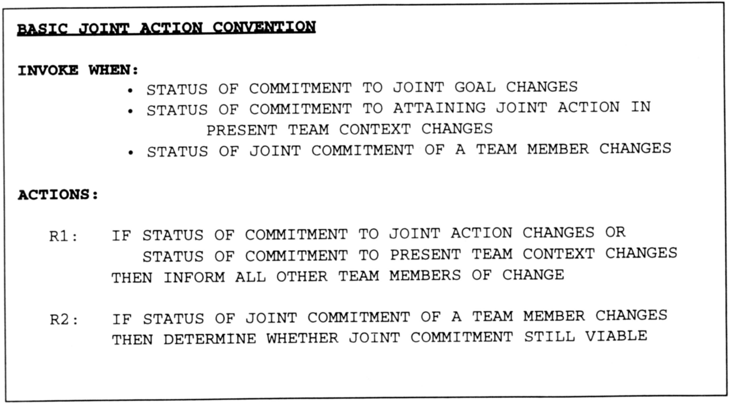


```
CONVENTION: Cohen and Levesque Model

REASONS FOR RE-ASSESSING COMMITMENT:
            • COMMITMENT SATISFIED
            • COMMITMENT UNATTAINABLE
            • MOTIVATION FOR COMMITMENT NO LONGER PRESENT

ACTIONS:

   R1:  IF   COMMITMENT SATISFIED OR
             COMMITMENT UNATTAINABLE OR
             MOTIVATION FOR COMMITMENT NO LONGER PRESENT
        THEN DROP COMMITMENT
```

(figure from (O’Hare & Jennings 1996))

Commitments and Conventions (Cont'd)

- **Example 2:**



BASIC JOINT ACTION CONVENTION

INVOKE WHEN:

- STATUS OF COMMITMENT TO JOINT GOAL CHANGES
- STATUS OF COMMITMENT TO ATTAINING JOINT ACTION IN PRESENT TEAM CONTEXT CHANGES
- STATUS OF JOINT COMMITMENT OF A TEAM MEMBER CHANGES

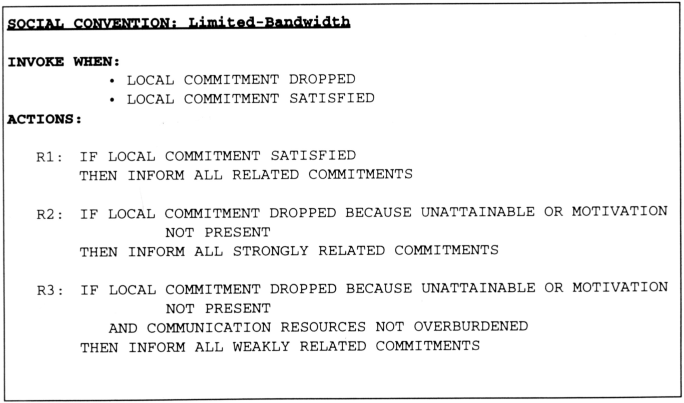
ACTIONS:

```
R1:   IF STATUS OF COMMITMENT TO JOINT ACTION CHANGES OR
         STATUS OF COMMITMENT TO PRESENT TEAM CONTEXT CHANGES
      THEN INFORM ALL OTHER TEAM MEMBERS OF CHANGE

R2:   IF STATUS OF JOINT COMMITMENT OF A TEAM MEMBER CHANGES
      THEN DETERMINE WHETHER JOINT COMMITMENT STILL VIABLE
```

(figure from (O'Hare & Jennings 1996))

Commitments and Conventions (Cont'd)

- **Example 3:**



```
SOCIAL CONVENTION: Limited-Bandwidth

INVOKE WHEN:
            • LOCAL COMMITMENT DROPPED
            • LOCAL COMMITMENT SATISFIED
ACTIONS:

   R1:  IF LOCAL COMMITMENT SATISFIED
        THEN INFORM ALL RELATED COMMITMENTS

   R2:  IF LOCAL COMMITMENT DROPPED BECAUSE UNATTAINABLE OR MOTIVATION
                 NOT PRESENT
        THEN INFORM ALL STRONGLY RELATED COMMITMENTS

   R3:  IF LOCAL COMMITMENT DROPPED BECAUSE UNATTAINABLE OR MOTIVATION
                 NOT PRESENT
           AND COMMUNICATION RESOURCES NOT OVERBURDENED
        THEN INFORM ALL WEAKLY RELATED COMMITMENTS
```

(figure from (O'Hare & Jennings 1996))