Roadmap – Course Overview

- Introduction
- Cooperation Models and Mechanisms

- Communication
- Micro-Level Architectures
- Macro-Level Structures
- Development Methods
Communication

- Basic Communication Models
  - Address Mode, Blocking, Buffering,
  - Communication Forms (message-/task-oriented)
- Human Communication
- Speech Act Theory
- Agent Communication Languages
- Interaction Protocols
- Ontologies
## Standard classification schema:

<table>
<thead>
<tr>
<th>Kriterium</th>
<th>Optionen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adressierung</td>
<td>direkt</td>
</tr>
<tr>
<td>Blockierung</td>
<td>synchron</td>
</tr>
<tr>
<td>Pufferung</td>
<td>ungepuffert</td>
</tr>
<tr>
<td>Kommunikationsform</td>
<td>meldungsorientiert</td>
</tr>
</tbody>
</table>

(table from (Weber 1998))
Direct addressing:

- sender and receiver communicate directly (point-to-point)
- symmetric: both sender and receiver name each other
  \[ Q: \text{send}(P, \text{message}) ; P: \text{receive}(Q, \text{message}) \]
- asymmetric: sender (client) names server, receiver (client) gets message only
  \[ Q: \text{send}(P, \text{message}) ; P: \text{receive}(\text{message}) \]

Indirect addressing:

- mailbox
- ports (special I/O points, often provided by operating system)
Blocking

Communication • Basic Communication Models

- **synchronous communication:**

  (figure from (Weber 1998))
asynchronous communication:

(figure from (Weber 1998))
Buffering

- **non-buffered communication**
  
  `receive`-command provides memory space (data structure), operating system writes incoming message into this space

- **buffered communication**
  
  if a receiver is not able to take messages, the operating system kernel saves these messages
Communication Forms

- Message-oriented communication
  
  sender posts message and expects (i) no reply or (ii) an acknowledgement of receipt

- task-oriented communication
  
  sender posts task specification, receiver replies with result of task execution

- both forms may occur in asynchronous and synchronous mode:

<table>
<thead>
<tr>
<th></th>
<th>asynchron</th>
<th>synchron</th>
</tr>
</thead>
<tbody>
<tr>
<td>meldungsorientiert</td>
<td>Datagramm</td>
<td>Rendezvous</td>
</tr>
<tr>
<td>auftragsorientiert</td>
<td>asynchroner entfernter Dienstaufruf</td>
<td>synchroner entfernter Dienstaufruf</td>
</tr>
</tbody>
</table>

(figure from (Weber 1998))
Communication Forms (Cont’d)

(figure from (Weber 1998))
Communication Forms (Cont’d)

Rendezvous

Klient

Server

send

Meldung

receive

Blockierung

(figure from (Weber 1998))
Communication Forms (Cont’d)

(figure from (Weber 1998))
Communication Forms (Cont’d)

(figure from (Weber 1998))
A note on terminology:

- synchroner entfernter Dienstaufruf = Remote-Procedure-Call (RPC)
- asynchroner entfernter Dienstaufruf = Remote-Service-Invocation (RSI)
- Datagramm = no-wait-send

RSI, in contrast to RPC: sender explicitly awaits result by means of receive
Roadmap – Chapter Details

Communication

- Basic Communication Models
- Human Communication
  - Richness
  - Complexity
- Speech Act Theory
- Agent Communication Languages
- Interaction Protocols
- Ontologies
Richness

- verbal vs non-verbal
  - miming, gestures, intonation, accent, ...
- intentional
- emotional
- a means for coordination
- speaking = physical activity + transfer of information (knowledge and belief)
Psychological complexity is a measure of the complexity of a message from a psychological perspective. Schultz von Thun (1996) identifies the following psychological dimensions of a message:

- **Sachinhalt**: The content or substance of the message.
- **Appell**: The appeal or call to action.
- **Nachricht**: The message or information.
- **Selbstoffenbarung**: The self-disclosure or self-revelation.
- **Beziehung**: The relationship or interaction.

These dimensions help to understand the complexity of human communication from a psychological standpoint.
Example (Schultz von Thun 1996):
Example (Cont’d):

Ampel ist auf Grün

Gib Gas

Da vorn ist Grün!

Ich hab es eilig

Du brauchst meine Hilfestellung
Communication

- Basic Communication Models
- Human Communication
- Speech Act Theory
  - Ingredients of a Speech Act
  - Performatives
- Agent Communication Languages
- Interaction Protocols
- Ontologies
Ingredients of a Speech Act

- Austin 1962, Searle 1970
- viewing messages as *actions*
- a speech act consists of:
  - Locution (physical utterance),
  - Illocution (intended meaning) and
  - Perlocution (resulting action).
- Example: “I am cold.”
  - ambiguity
Use *performatives* to distinguish illocutionary force: *promise, report, convince, insist, request, demand*, etc.

Categories of performatives:
1. assertives,
2. directives,
3. commissives,
4. declaratives,
5. expressives
**Performatives (Cont’d)**

Commonly used performatives:

<table>
<thead>
<tr>
<th>Performative</th>
<th>Messg. Type</th>
<th>Illoc. Force</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>inform</td>
<td>Assertion</td>
<td>Declarative</td>
<td>belief revision</td>
</tr>
<tr>
<td>ask</td>
<td>Query</td>
<td>Directive</td>
<td>reply</td>
</tr>
<tr>
<td>reply</td>
<td>Assertion</td>
<td>Assertive</td>
<td>acceptance</td>
</tr>
<tr>
<td>request</td>
<td>Query</td>
<td>Directive</td>
<td>action/information</td>
</tr>
<tr>
<td>command</td>
<td>Assertion</td>
<td>Directive</td>
<td>action execution</td>
</tr>
<tr>
<td>allow</td>
<td>Assertion</td>
<td>Directive</td>
<td>acceptance</td>
</tr>
<tr>
<td>propose</td>
<td>Query</td>
<td>Assertive</td>
<td>counter-proposal?</td>
</tr>
<tr>
<td>confirm</td>
<td>Assertion</td>
<td>Commissive</td>
<td>acceptance</td>
</tr>
<tr>
<td>prefer</td>
<td>Assertion</td>
<td>Expressive</td>
<td>belief revision?</td>
</tr>
</tbody>
</table>
Communication

- Basic Communication Models
- Human Communication
- Speech Act Theory
- Agent Communication Languages (ACLs)
  - KQML
  - KIF
- Interaction Protocols
- Ontologies
KQML = Knowledge Query and Manipulation Language (Labrou & Finin 1994)

Message format:

(performative
  :sender <word>
  :receiver <word>
  :in-reply-to <word>
  :reply-with <word>
  :language <word>
  :ontology <word>
  :content <expression>)

Various kinds of performatives
Example:

(advertise
 :sender Agent1
 :receiver Agent2
 :in-reply-to ID1
 :reply-with ID2
 :language KQML
 :ontology kqml-ontology
 :content (ask
 :sender Agent1
 :receiver Agent3
 :language Prolog
 :ontology blocks-world
 :content "on(X,Y)")
)
KIF = Knowledge Interchange Format

KQML does not say anything about content of messages → need content languages/ontologies

KIF is a logical language to describe contents/ knowledge (first-order logic with some extensions/restrictions)

Examples of KIF formulae:

- \((\text{=> } (\text{and} (\text{real-num } ?x) (\text{even-num } ?n)) (> (\text{expt } ?x ?n) 0)))\)
- \((\text{interested joe '(salary ,?x ,?y ,?z)})\)
Communication

- Basic Communication Models
- Human Communication
- Speech Act Theory
- Agent Communication Languages (ACLs)
- Interaction Protocols
  - What is an (Interaction) Protocol?
  - Types of Interaction Protocols
  - Designing Interaction Protocols
  - Protocol Examples
- Ontologies
What is an (Interaction) Protocol?

Definition:

- “[an interaction protocol is] an interaction regime that guides the agents” (Koning, Francois & Demazeau 1999)

- Restrict the range and ordering of possible messages

- Usually formalized by state diagrams or *Interaction Diagrams* in FIPA-AgentUML
  (FIPA = Foundation for Intelligent Physical Agents)
Types of Interaction Protocols

For every type of interaction, there is a protocol!

- Argumentation protocols
- Contracting protocols
- Auctions protocols
- Bargaining protocols
- Voting protocols
- Brokering protocols
- Matchmaking protocols
- Authentication protocols
Six-step process (Koning, Francois & Demazeau 1999):

1. describe the interaction capabilities of the agents in use,
2. clarify the types of messages involved,
3. describe the agents’ behaviours,
4. explain the possible message sequences between agents,
5. clarify the various internal agent states,
6. establish the diagram of the protocol.)
Examples (1) – A Basic Protocol
Examples (2) – The Contract Net Protocol

Communication • Interaction Protocols
Examples (3) – A Brokering Protocol

Communication • Interaction Protocols

FIPA-Brokering-Protocol

Initiator

Broker

proxy (proxied-communicative-act, agent-description, proxy-condition)

not-understood

refuse

agree

failure-no-match

[agreed, cannot find any target agents]

failure-proxy

inform-done-proxy

[agreed, find any target agents]

failure-sub-protocol

inform-done-sub-protocol

inform-result-sub-protocol

inform-done-proxy, sub-protocol proceed

proposed-communicative-act

[agreed, find any target agents]

[action-condition]

reply-message

[sub-protocol proceed]
Examples (4) – English Auction Protocol
Communication

- Basic Communication Models
- Human Communication
- Speech Act Theory
- Agent Communication Languages (ACLs)
- Interaction Protocols
- Ontologies
  - What is an Ontology?
  - Examples
  - Principles for Ontology Design
What is an Ontology?

Ontology =

- description of relevant objects and relations in a domain
- a formal, explicit specification of a shared conceptualization (Gruber 1993)
  - conceptualization: abstract model of some phenomenon, identifies its relevant characteristics
  - explicit: model (characteristics) are formulated explicitly
  - formal: machine-readable
  - shared: captures consensual knowledge (i.e., accepted not only by a single individual)
Expressing ontologies in simple statements:

- (class Block), (class PhysicalObject),
  (subclassOf Block PhysicalObject)

- \( \forall x, y, z \) (instanceOf x y \land (subclassOf y z) \implies (instanceOf x z))

- (domain On-Table PhysicalObject)

- (range On-Table PhysicalObject)
Expressing ontologies in tree structures:

- **Entity**: Person, Enterprise
- **Agent**: Person, Employee
- **Attribute**: Agent Attribute (Age), Enterprise Attribute (Turnover)
- **Relation**: Employment Relation (works-for), reports-to

Diagram:
- **Thing**
  - **Entity**
    - **Agent**
      - **Person**
      - **Enterprise**
    - **Agent Attribute**
      - **Age**
  - **Attribute**
    - **Enterprise Attribute**
      - **Turnover**
  - **Relation**
    - **Employment Relation**
      - **works-for**
      - **reports-to**
Principles for Ontology Design

- **Clarity**: minimize ambiguity, motivate distinctions, give examples
- **Coherence**: internal consistency
- **Extendibility**: extension of existing terms without need to revise existing definitions
- **Minimal encoding bias**: ideally representation choices are not made for the convenience of notation or implementation
- **Minimal ontological commitment**: ontology should make as few claims as possible about the world being modelled (parties committed to the ontology are free to specialize and instantiate the ontology as needed)
there are not only many “ready-to-use” ontologies available already ...

- common sense ontologies, domain ontologies, task ontologies, etc.
- e.g. CYC, WordNet, PIF (business process modeling), PhysSys (knowledge about physical system processes), AIRCRAFT (air-campaign planning knowledge)

... but also many languages for ontology specification

- KIF, Ontolingua, Frame Logic, CLASSIC, LOOM, CycL, etc.
- languages being conform with Semantic Web standards: SHOE, XOL, OIL, DAML-OIL, etc.

... and many useful ontology design tools

- Protege, Webonto, ONTOEDIT, Ontobroker, etc.