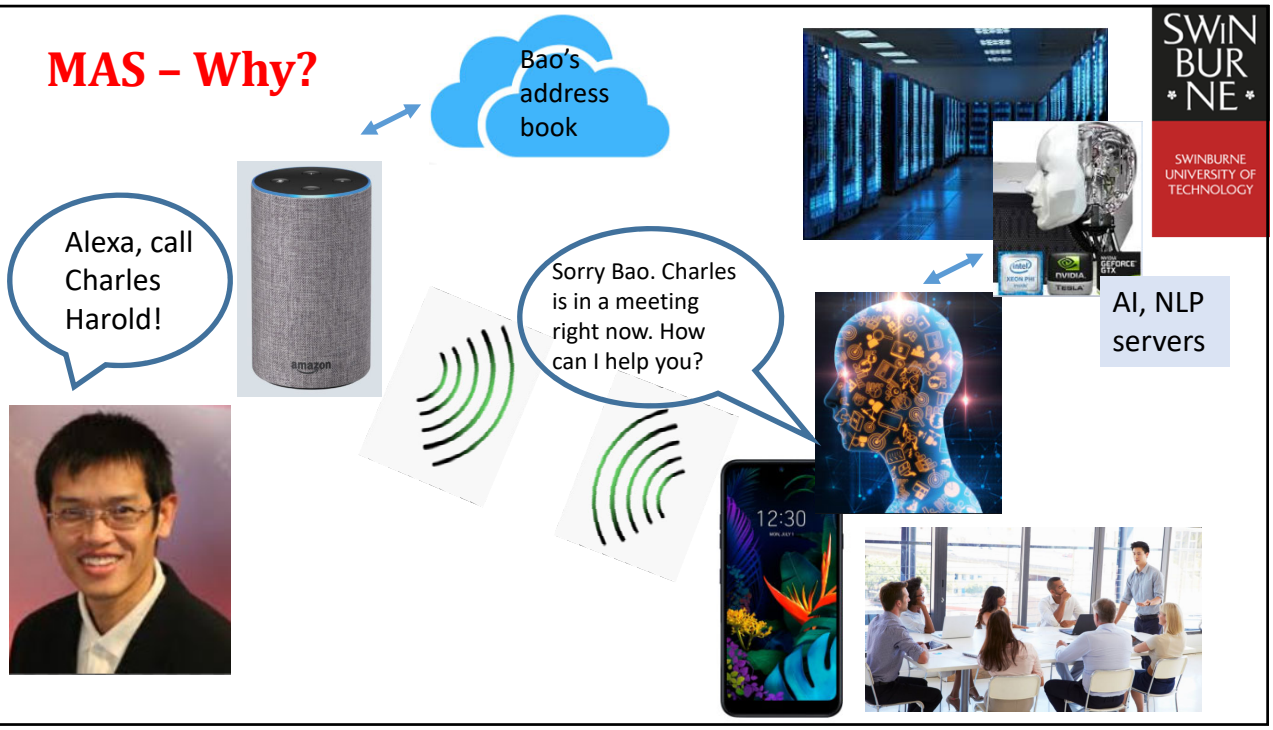
Outline

- Multiagent Systems (**MAS**)
  - Why?
  - What?
  - How?
- Agent communication
- Agent interactions
- Automated negotiation

MAS – Why?



MAS – Why?

- Computer systems today no longer stand alone, but are networked into large distributed systems
- The Internet is an obvious example, but networking is spreading its ever-growing tentacles…
  - The Internet of Things (IoT)
- As interconnected systems and subsystems have become smarter
  - They increasingly display the behaviours as intelligent agents who communicate, interact, coordinate, cooperate, compete, etc.

MAS – Why?

- Computer systems today no longer stand alone, but are networked into large distributed systems
- The Internet is an obvious example, but networking is spreading its ever-growing tentacles…
  - The Internet of Things (IoT)
- As interconnected systems and subsystems have become smarter
  - They increasingly display the behaviours as intelligent agents who communicate, interact, coordinate, cooperate, compete, etc.

MAS – What?

- A multiagent system is one that consists of a number of agents, which *interact* with one-another
- In the most general case, agents will be acting on behalf of users with different goals and motivations
- To successfully interact, they will require the ability to *cooperate*, *coordinate*, and *negotiate* with each other, much as people do

MAS – Two key problems

- How do we build agents capable of independent, autonomous action, so that they can successfully carry out tasks we delegate to them?
- How do we build agents that are capable of interacting (cooperating, coordinating, negotiating) with other agents in order to successfully carry out those delegated tasks, especially when the other agents cannot be assumed to share the same interests/goals?
- The first problem is *agent design*, the second is *society design* (micro/macro)

MAS – Specific questions to consider

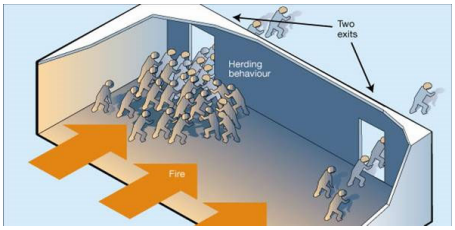
- In Multiagent Systems, we address issues such as:
  - How can cooperation emerge in societies of self-interested agents?
  - What kinds of languages can agents use to communicate?
  - How can self-interested agents recognize conflict, and how can they (nevertheless) reach agreement?
  - How can autonomous agents coordinate their activities so as to cooperatively achieve goals?

Multi-agent communication & coordination

People participating in an auction (Competitive behaviour)

Source: https://www.4for4.com/

Building egress (Coordinated behaviour)



Source: Coordination of multi-agent systems, Mark Spong

Mobile Robot Networks

Multi-agent communication & coordination

Flock of birds

Source: https://www.howitworksdaily.com/

Coalition Formation

Source: http://on9groupbuying.blogspot.com/

Agent Communication

- Macro-aspects of intelligent agent technology i.e. those related to the agent society, rather than the individual
  - *Communication* - request cooperation
  - *Cooperation* – the act of working together to achieve some common goal

How do agents communicate?

Speech Acts

- Most treatments of communication in (multi-) agent systems borrow their inspiration from *speech act theory*
- Speech act theories are *pragmatic* theories of language, i.e., theories of language use: they attempt to account for how language is used by people every day to achieve their goals and intentions
- The origin of speech act theories are usually traced to Austin's 1962 book, *How to Do Things with Words*

8-12

Speech Acts

- Austin noticed that some utterances are rather like 'physical actions' that appear to *change the state of the world*
- Paradigm examples would be:
  - declaring war
  - christening
  - 'I now pronounce you man and wife' :-)
- But more generally, *everything* we utter is uttered with the intention of satisfying some goal or intention
- A theory of how utterances are used to achieve intentions is a speech act theory

8-13

Speech Acts

- Searle (1969) identified various different types of speech act:
  - *representatives:*
    such as *informing*, e.g., 'It is raining'
  - *directives:*
    attempts to get the hearer to do something e.g., 'please make the tea'
  - *commisives:*
    which commit the speaker to doing something, e.g., 'I promise to… '
  - *expressives:*
    whereby a speaker expresses a mental state, e.g., 'thank you!'
  - *declarations:*
    such as declaring war or christening

8-14

Speech Acts

- There is some debate about whether this (or any!) typology of speech acts is appropriate
- In general, a speech act can be seen to have two components:
  - a *performative verb:*
    (e.g., request, inform, promise, …)
  - *propositional content:*
    (e.g., "the door is closed")

8-15

Speech Acts

- Consider:
  - performative = request
    content = "the door is closed"
    speech act = "please close the door"
  - performative = inform
    content = "the door is closed"
    speech act = "the door is closed!"
  - performative = inquire
    content = "the door is closed"
    speech act = "is the door closed?"

8-16

Plan Based Semantics

- How does one define the semantics of speech acts? When can one say someone has uttered, e.g., a request or an inform?
- Cohen & Perrault (1979) defined semantics of speech acts using the *precondition-delete-add* list formalism of planning research
- Note that a speaker cannot (generally) *force* a hearer to accept some desired mental state

8-17

Plan-Based Semantics

- Here is their semantics for *request*:
  request(s, h, ϕ)

  pre:
  - *s* believe *h* can do ϕ
    (you don't ask someone to do something unless you think they can do it)
  - *s* believe *h* believe *h* can do ϕ
    (you don't ask someone unless *they* believe they can do it)
  - *s* believe *s* want ϕ
    (you don't ask someone unless you want it!)

  post:
  - *h* believe *s* believe *s* want ϕ
    (the effect is to make them aware of your desire)

8-18

KQML and KIF

- We now consider *agent communication languages* (ACLs) — standard formats for the exchange of messages
- The best known ACL is KQML, developed by the ARPA knowledge sharing initiative
- KQML is comprised of two parts:
  - the knowledge query and manipulation language (KQML)
  - the knowledge interchange format (KIF)

8-19

KQML and KIF

- KQML is an 'outer' language, that defines various acceptable 'communicative verbs', or *performatives* Example performatives:
  - `ask-if` ('is it true that. . . ')
  - `perform` ('please perform the following action. . . ')
  - `tell` ('it is true that. . . ')
  - `reply` ('the answer is . . . ')
- KIF is a language for expressing message *content*

8-20

KIF – Knowledge Interchange Format

Used to state:

- Properties of things in a domain (e.g., “Bao is a lecturer”)
- Relationships between things in a domain (e.g., “Sam is enrolled in COS30018”)
- General properties of a domain (e.g., “All students are enrolled in at least one course”)

8-21

KIF – Knowledge Interchange Format

- “The temperature of Hawthorn is 17 Celsius”:

```
(= (temperature Hawthorn) (scalar 17 Celsius))
```

- “An object is a bachelor if the object is a man and is not married”:

```
(defrelation bachelor (?x) :=
    (and (man ?x) (not (married ?x))))
```

- “Any individual with the property of being a person also has the property of being a mammal”:

```
(defrelation person (?x) :=> (mammal ?x))
```

8-22

KQML and KIF

- In order to be able to communicate, agents must have agreed on a common set of terms
- A formal specification of a set of terms is known as an *ontology*
- The knowledge sharing effort has associated with it a large effort at defining common ontologies — software tools like ontolingua for this purpose
- Example KQML/KIF dialogue…

```
A to B: (ask-if (> (size chip1) (size chip2)))
B to A: (reply true)
B to A: (inform (= (size chip1) 20))
B to A: (inform (= (size chip2) 18))
```

8-23

FIPA - ACL

- More recently, the Foundation for Intelligent Physical Agents (FIPA) started work on a program of agent standards — the centerpiece is an ACL
- Basic structure is quite similar to KQML:
  - *performative*
    20 performative in FIPA
  - *housekeeping*
    e.g., sender, etc.
  - *content*
    the actual content of the message

8-24

FIPA - ACL

- Example:

```
(inform
   :sender       agent1
   :receiver agent5
   :content  (price good200 150)
   :language sl
   :ontology hpl-auction
)
```


FIPA - ACL

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

Interaction Protocols

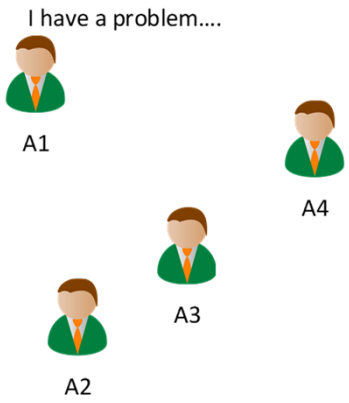
- ACL defines the syntax and semantics of individual utterances
- ***Interaction protocols*** specify the rules of interaction
  - Rules of procedure for the conversation, including:
    - The permissible agent roles involved in the interaction
    - Sequence of message exchanges between agent roles, and
    - Constraints on the content of messages exchanged
- All participating agents must conform to the interaction protocol to enable rational conversations

Example: The Contract Net Protocol

- Contract Net Protocol (CNP) is a well known interaction protocol for task allocation
- It comprises 5 stages
  - Recognition
  - Announcement
  - Bidding
  - Awarding
  - Expediting

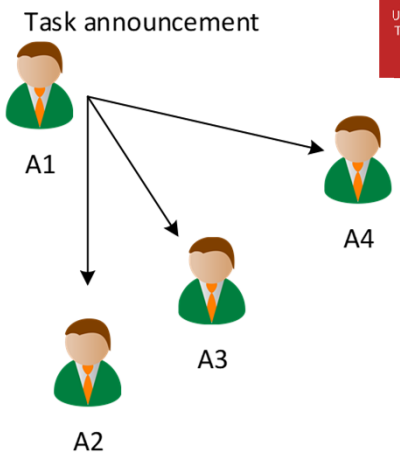
1. Recognition

- In this stage, an agent recognizes that it has a problem that it wants help because
  - It cannot solve the problem in isolation, or
  - It would prefer not to solve it in isolation (because of solution quality, deadline constraints etc.)



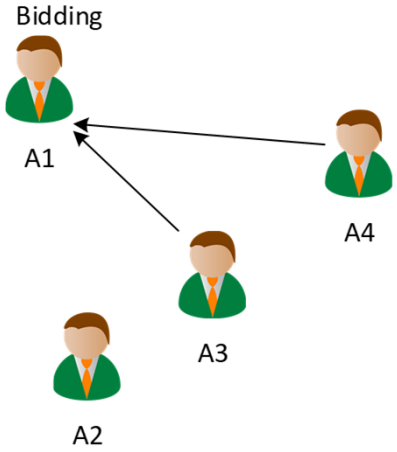
2. Announcement

- In this stage, the agent sends out (broadcasts) an announcement with a specification of the task to be achieved.
- Specification must encode
  - Description of the task
  - Any constraints (e.g. deadlines, quality constraints)
  - Meta-task information (e.g. "bids must be submitted by…")



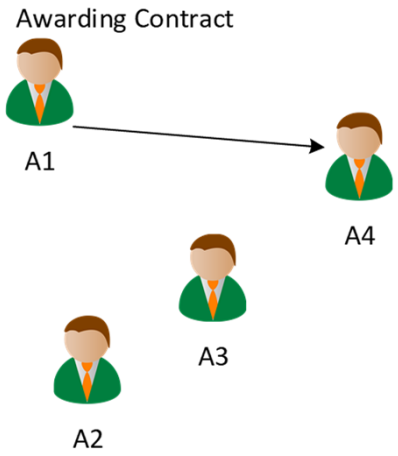
3. Bidding

- Agents receiving the announcement decide whether they want to *bid* for the task
- Factors affecting the decision include:
  - Is the agent capable of expediting the task
  - What are the relevant quality constraints and price information (if relevant)
- If an agent chooses to bid, then it submits a *tender*



4. Awarding & Expediting

- Task owner must decide who to award the contract to
- The result of the decision is communicated to the winning agent
- The losing agents are also notified
- The successful *contractor* expedites the task
- May involve generating further manager-contractor relationships: *sub-contracting*



Designing Interaction Protocols

- *Top-down approach* – IP specifications are publicly available
  - E.g. FIPA Contract Net Protocol and Iterated Contract Net Protocol
  - Agent designer can implement agent to play the appropriate role in the interaction
  - Benefit – agent can engage in *meaningful conversations* by following the pre-defined IP
- *Bottom-up approach* - Agent designer makes the agents sufficiently aware of the meanings of the messages, and the goals, beliefs and other mental attitudes that they possess
  - Agent's planning process causes IPs to arise spontaneously from the agents' choices

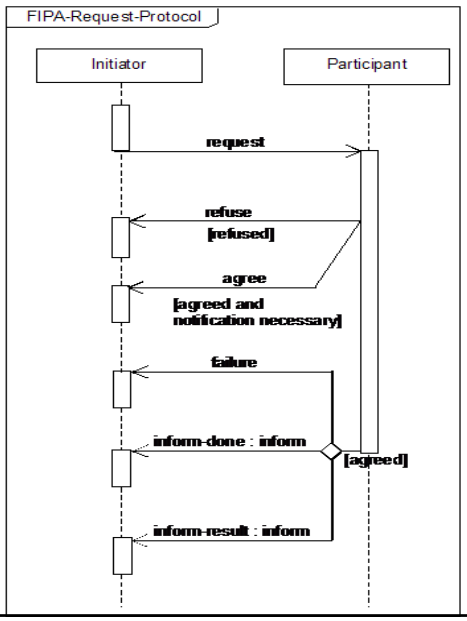
FIPA Interaction Protocols

- FIPA Request Interaction Protocol
- FIPA Query Interaction Protocol
- FIPA Request When Interaction Protocol
- FIPA Contract Net Interaction Protocol
- FIPA Iterated Contract Net Interaction Protocol
- FIPA Brokering Interaction Protocol
- FIPA Subscribe Interaction Protocol
- FIPA Propose Interaction Protocol
- *See:* http://www.fipa.org/repository/ips.html

FIPA Request Interaction Protocol

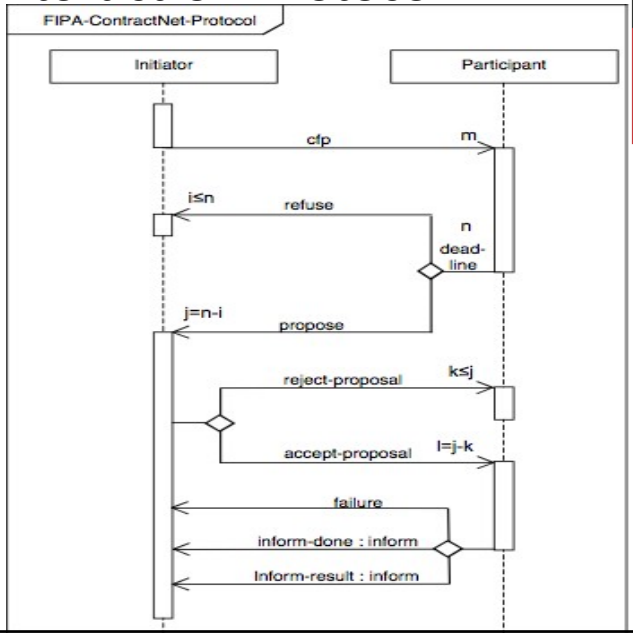
- Allows one agent to request another to perform some action

http://fipa.org/specs/fipa00026/SC00026H.html



FIPA Contract Net Interaction Protocol

http://fipa.org/specs/fipa00029/SC00029H.html

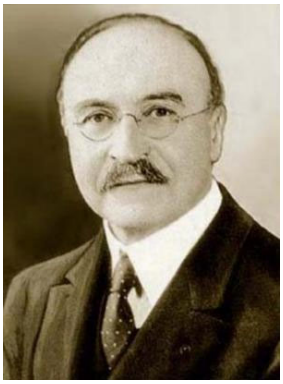


SWINBURNE UNIVERSITY OF TECHNOLOGY

Presentation Outline

- Automated Negotiation
- Types of negotiation
- Negotiation Elements
- Bilateral Negotiation
  - Utility function, Concessions, Trade offs
  - Negotiation tactics
- Learning & Reasoning in Negotiation
- Automated Negotiating Agent's Competition (ANAC)
- GENIUS Environment

Baekeland & Kodak



Leo Hendrik Baekeland
Invented a type of photographic paper (1891)

In 1899, Eastman Kodak sent an invitation to Baekeland to discuss the sale of his Velox photosensitive manufacturing business

Baekeland wanted to ask for **$50,000** but wasn't sure if he could even get **$25,000**

Eastman spoke first and made an offer of **$1,000,000**

Baekeland accepted the offer ☺

He used his share of the profit for research and created plastic (Bakelite)

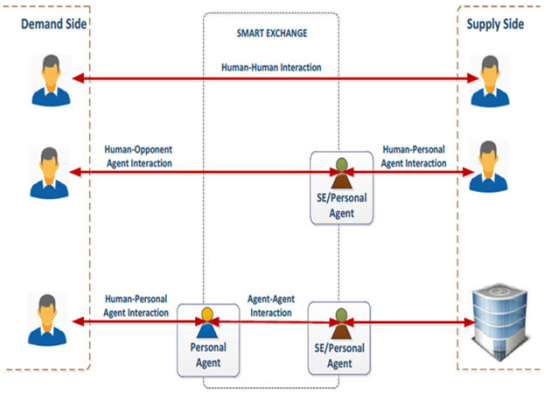
George Eastman
- Founded Eastman Kodak Company (1888)
- Popularised the use of roll film

Negotiation

- Core human activity – to form alliances, to reach trade agreements, to resolve conflicts in business and in personal life
  - Job negotiation, acquiring a house, haggling at the marketplace, negotiating assignment submission deadline
- Important topic of research in economics, AI, game theory and social psychology
- The focus in the last two decades has been on automation of negotiation and e-negotiation systems

Automated Negotiation

- Can we delegate negotiation to automated agents?
- Can software agents find better outcomes than humans?
- Automation leads to reduced negotiation time and costs
- Users can avoid social confrontation
- Possibility to find better deals by exploration of the outcome space
- Improve negotiation skills of human negotiator

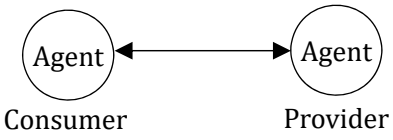


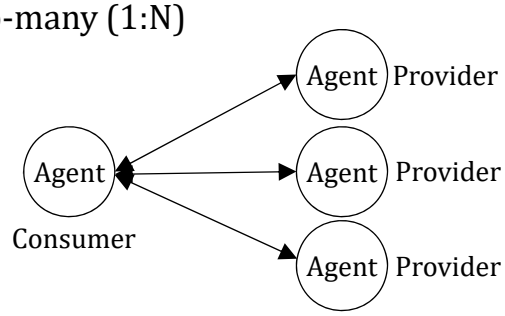
Automated Negotiation

- Form of interaction by which two (or more) parties (agents) try to reach a mutually acceptable agreement on some matter
  - The agents' interests are possibly conflicting requiring them to resolve the differences
  - Process of joint-decision making
- Automated Negotiation – negotiation where at least one of the participants is a software agent

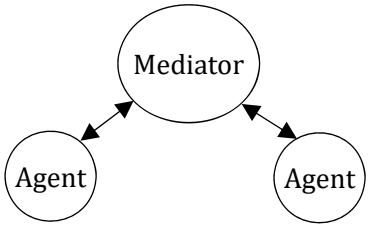
Types of Negotiation

- One-to-one (1:1) vs. One-to-many (1:N)





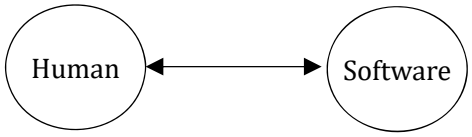
- Direct vs Mediated



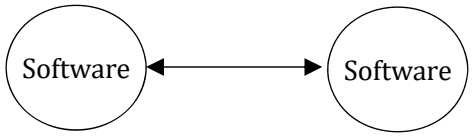
- Single-Issue vs. Multi-Issue (aka. multi-attribute)
  - Price of a car
  - Price, warranty, accessories, capped price servicing

Types of Negotiation

- Human vs Software Agent

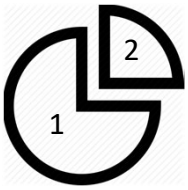


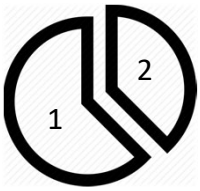
- Software vs Software Agent



Distributive Negotiation

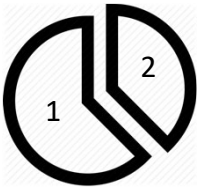
- Zero-sum or win-lose
- Competitive
- Typically single-issue negotiation
- "Fixed pie" – who gets the bigger share (individual gain)
- Example – used car sale
  - Salesman wants to sell at the highest price possible
  - Buyer wants to pay the least amount possible

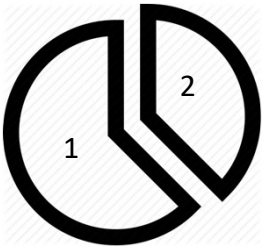




Integrative Negotiation

- Win-win solution possible
- Aiming to maximise cooperativeness
- Potential for negotiating parties' interests to be combined in different ways to *enlarge the pie*
- Multi-issue negotiation
- **Example:** Buying a laptop with the standard 2-year warranty: Seller wants $2,000 as his cost is already $1,900 and buyer only wants to pay $1,800. We can "**enlarge**" the pie by considering the 5-year extended warranty which is valued by the buyer at $400 while it only costs the seller $50. Both parties are happy to reach the agreement at $2,100 for the laptop with the 5-year warranty!



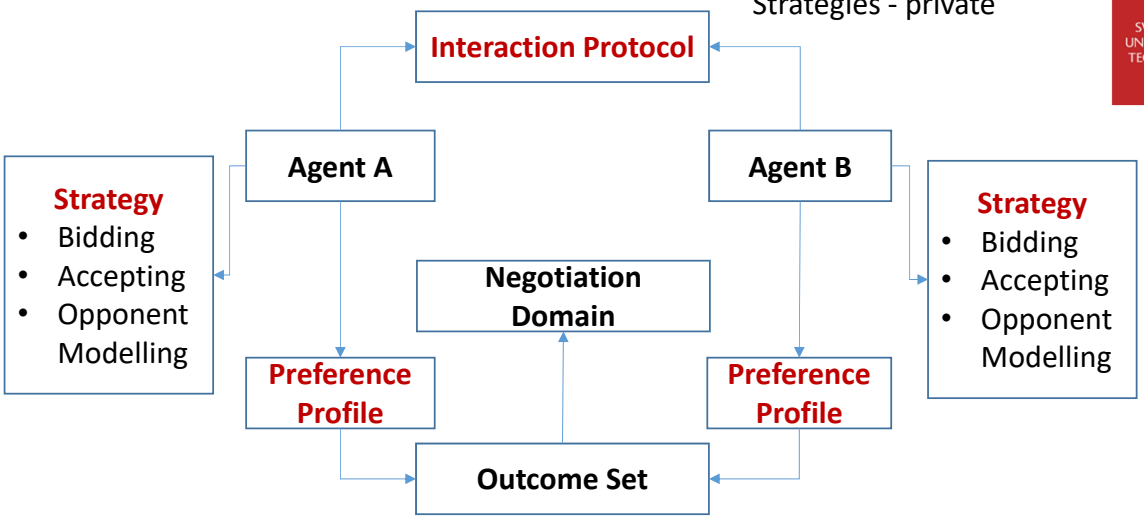


Pareto optimality/Pareto Frontier

- A *Pareto optimal* outcome is one where no agent can increase its utility without reducing the other agent's utility
- If an outcome is not Pareto optimal, then there is potential, through re-negotiation, for at least one of the agents to get a better outcome without reducing the value of the other
- The line that connects all Pareto optimal agreements is called the *Pareto frontier*
- A perfectly executed integrative negotiation results in Pareto optimality
- Negotiating agents are not aware of the Pareto frontier since they are not aware of the opponent utility of the offers/counter offers

Key Negotiation Elements

Preferences – private
Protocols – public
Strategies - private



Preferences & Utility function

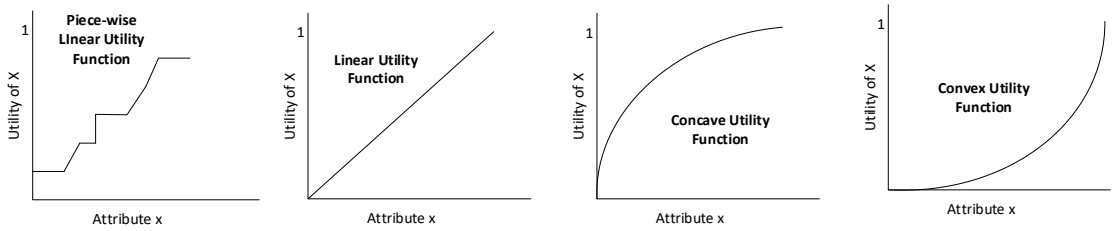
- **Utility** – a measure of the relative satisfaction from or desirability of consumption of a product or service
- Two types of utility
  - **Ordinal utility** – allows ranking of alternatives
    - blue $\succ$ green $\succ$ yellow
  - **Cardinal Utility** - allows measurement of the strength of preference of a good or service with precision through the use of some objective criteria
    - U(blue) = 0.8
    - U(green) = 0.7
    - U(yellow) = 0.65

Preferences & Utility function

- Utility Function is a mathematical function that can assign an utility value to every single alternative outcome
- Cardinal utility function → can assign a scalar value to every possible value assignment

$$U : C \rightarrow [0,1]$$

where C is the set of all possible assignments



Multi-attribute Utility

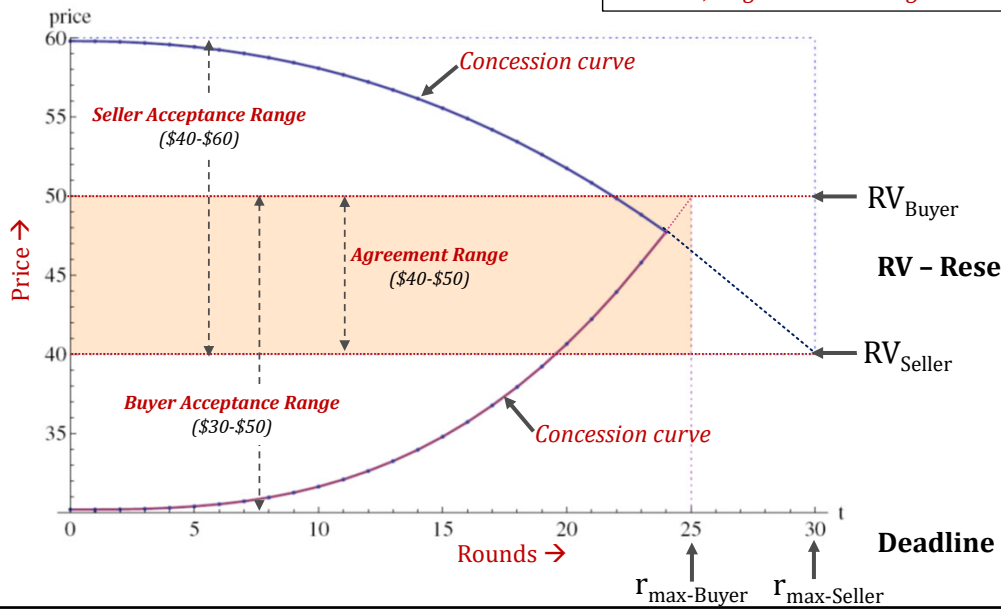
- When we consider *multi-attribute negotiation*, overall utility is a function of the individual utility functions
- Simplest way to express multi attribute utility is as a linear additive function

$$MAU(x_1 \ldots . xn) = \sum_{i=1}^{n} w_i . ui$$

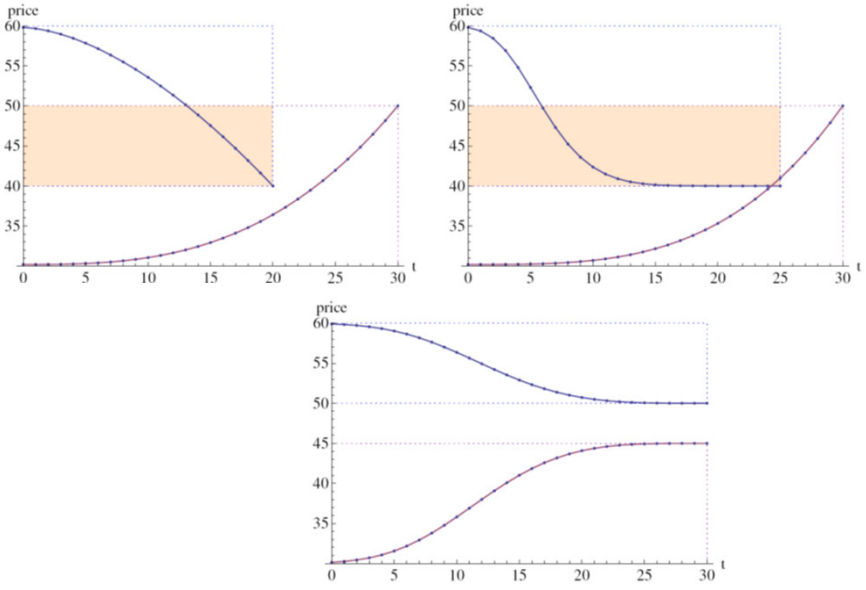
- Each attribute is assigned a weight (indication of importance)
- Each issue is evaluated separately

Automated Negotiation

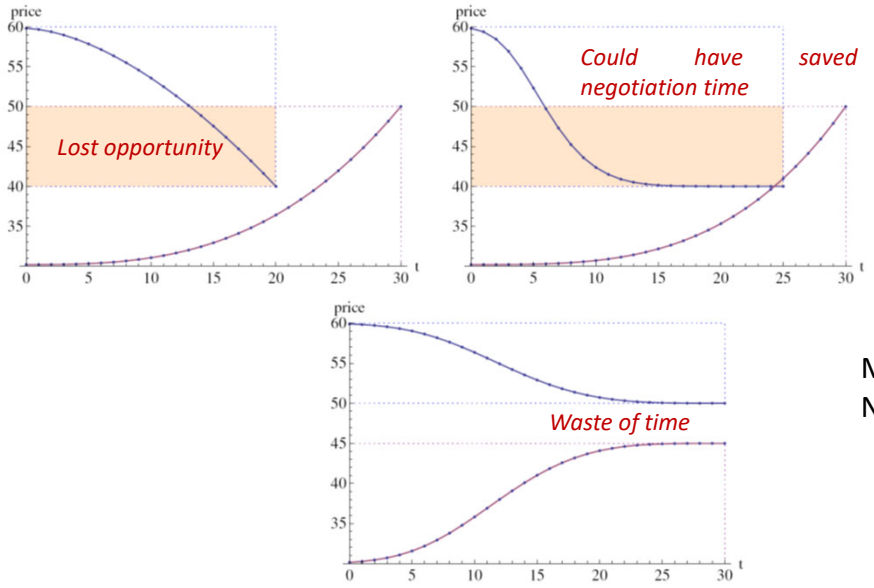
Buyer and seller negotiating over the price of a second-hand book – bilateral, single-attribute negotiation



Negotiation Examples



Negotiation Examples



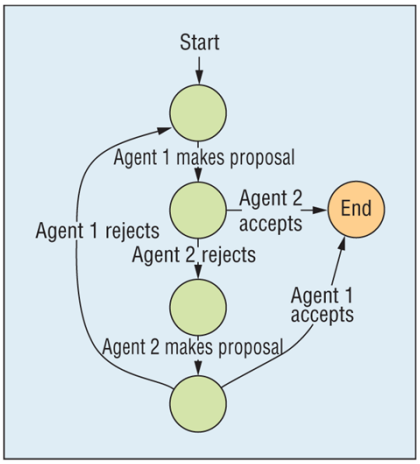
Mediated Negotiation?

Negotiation Protocols

- Set of rules that govern how negotiation takes place
- How many participants are allowed?
- What are the valid actions of participants?
- When does negotiation finish?
- Structure of possible agreements
  - Which messages are sent by whom, to whom, and at what stage

Alternating Offers protocol (One-to-one Negotiation)

- Negotiation between two agents
  - e.g. a service consumer and provider
- Single or multi-issue negotiation
  - Sequence of rounds
- May or may not have a deadline (public or private)
- Negotiation thread - Sequence of offers and counter-offers
- Tactics are used to generate offers and counter-offers
- Scoring functions are used to evaluate offers and counter-offers



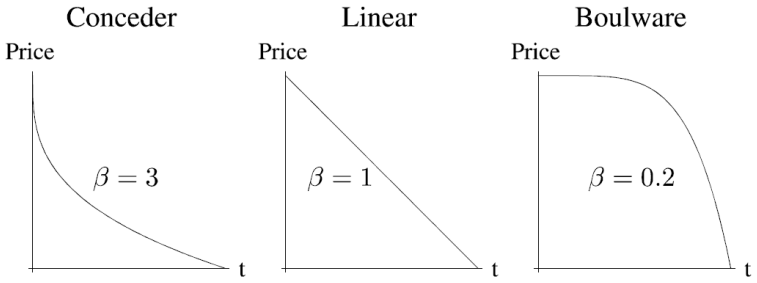
Source: Fatima, Shaheen, Sarit Kraus, and Michael Wooldridge. "The Negotiation Game." *IEEE Intelligent Systems* 29.5 (2014): 57-61.

Strategies/Tactics

- Agents use tactics to generate new values for each variable in a counter offer
- *Negotiation tactic* – a function that determines how to compute a new value for a negotiation issue by considering some criteria
- Some popular heuristics based tactics
  - *Time dependant* – negotiation deadline
  - *Resource dependant* – time, number of negotiating agents
  - *Behaviour dependant* – imitate the opponent's behaviour Tit-for-tat

Time-dependent Tactics

- Time (*deadline*) and *reservation value* are the two factors that decide what value to offer next



- *Conceder* – concedes very quickly to go close to reservation value
- *Boulware* – maintains the initially offered value until the time is almost exhausted

Resource-dependent Tactics

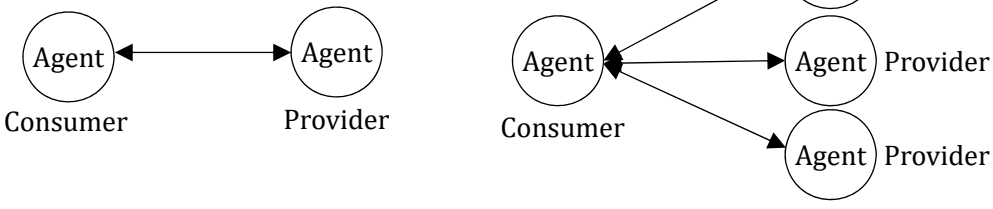
- Calculate offers based on amount of available resources e.g. number of agents
  - If negotiating with more agents, then less pressure to reach agreement
  - If resource is time (more time left, less pressure)
    - Is **Time Dependent** tactic a kind of **Resource Dependent** tactic??

Behaviour-dependent Tactics

- Compute next offer based on previous attitude of negotiation opponent (imitate opponent behaviour)
- Tit-for-tat (TFT) negotiation strategy posits that an agent can reach a successful agreement if it behaves cooperatively.
  - Implementing a tit-for-tat strategy occurs when one agent cooperates with another agent in the very first interaction and then mimics their subsequent moves. This strategy is based on the concepts of retaliation and altruism.

One-to-Many Negotiations

- One buyer negotiating with multiple sellers for the same service
  - Should the buyer negotiate separately with each agent?
  - Should the buyer make same offers/counter-offers to all agents?
  - What if they are using different strategies?



Automated Negotiation Challenges

- Dynamic and distributed environment
- Limited and uncertain information about
  - Negotiation partners
    - Difficult to anticipate partner's behaviour with limited knowledge
  - Private information – utility function, decision model, reservation value, deadline
  - Conflicting preferences with no knowledge about private parameters of the opponent
- Integrative negotiation not always possible
- Long negotiation times – especially if solution space is large

Learning and Reasoning in Negotiation

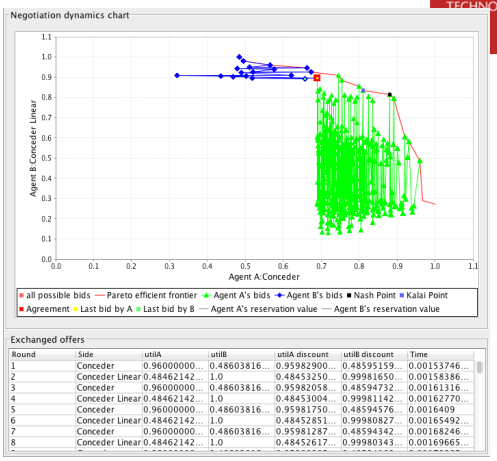
- Finding negotiation strategies that can anticipate opponents strategic parameters and adapt to opponent behaviour
- Approaches either rely on *prior knowledge* or assume *no prior knowledge*
  - *Bayesian Reasoning*
    - Beliefs about the environment and other partner's behavior are modelled explicitly with a probabilistic framework
  - *Case-based Reasoning*
  - *Neural Networks*
  - *Markov-Decision Processes (MDPs)*
  - *Reinforcement Learning*
  - *Evolutionary Computing*

Automated Negotiating Agents Competition (ANAC)

- Annual event held in conjunction with International Joint Conference on Artificial Intelligence (IJCAI)
  - Previously with International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)
- Main goals of the competition include:
  - To encourage the design of practical negotiation agents that can negotiate against unknown opponents and in a variety of circumstances
  - To provide a benchmark for objectively evaluating different negotiation strategies
  - To explore different learning and adaptation strategies and opponent models, and
  - To collect state-of-the-art negotiating agents and negotiation scenarios and make them available to the wider community

GENIUS

- *Genius* – General environment for Negotiation with Intelligent multi-purpose Simulation (http://ii.tudelft.nl/genius/)
- Testbed for negotiating agents that includes:
  - Set of negotiation problems
  - Library of negotiation strategies *boulware, conceder, linear etc.*
  - Analytical tools for evaluating agent performance



GENIUS

- User guide
- Source code
- Number of different negotiation strategies implemented
  - Time dependent (boulware/conceder)
  - Tit-for-tat
  - Other strategies from the ANAC competition
- Code snippets to show how you can build your own negotiation agent

- https://medium.com/@malintha/writing-a-simple-agent-in-genius-43f13b186e5

Summary

- What are Multiagent Systems (MAS)? And how to design them?
- Agent communication
- Agent interaction
  - Languages
  - Protocols
- Automated negotiation