The aim is to give an overview of the ways that theorists conceptualise agents, and to summarise some of the key developments in agent theory.

- Begin by answering the question: *why theory?*
- Discuss the various different attitudes that may be used to characterise agents.
- Introduce some problems associated with formalising attitudes.
- Introduce modal logic as a tool for reasoning about attitudes, focusing on knowledge/belief.
- Introduce Epistemic Logic for multi-agent systems.
- Introduce BDI logic.
Formal methods have (arguably) had little impact of general practise of software development: why should they be relevant in agent based systems?

The answer is that we need to be able to give a semantics to the architectures, languages, and tools that we use – literally, a meaning.

Without such a semantics, it is never clear exactly what is happening, or why it works.

End users (e.g., programmers) need never read or understand these semantics, but progress cannot be made in language development until these semantics exist.

In agent-based systems, we have a bag of concepts and tools, which are intuitively easy to understand (by means of metaphor and analogy), and have obvious potential.

But we need theory to reach any kind of profound understanding of these tools.

Where do theorists start from?

The notion of an agent as an intentional system . . .

So agent theorists start with the (strong) view of agents as intentional systems: one whose simplest consistent description requires the intentional stance.
We want to be able to design and build computer systems in terms of ‘mentalistic’ notions.
Before we can do this, we need to identify a tractable subset of these attitudes, and a model of how they interact to generate system behaviour.
So first, which attitudes?
Two categories:
- Information attitudes
  - belief
  - knowledge
- pro-attitudes
  - desire
  - intention
  - obligation
  - commitment
  - choice
  - ...

So how do we formalise attitudes?
Consider...
Janine believes Cronos is father of Zeus.
Naive translation into first-order logic:
\[ \text{Bel}(\text{Janine}, \text{Father(Zeus, Cronos)}) \]
But...
- the second argument to the \text{Bel} predicate is a formula of first-order logic, not a term;
- need to be able to apply \text{Bel} to formulae;
- allows us to substitute terms with the same denotation: consider \text{Zeus} = \text{Jupiter}

intentional notions are referentially opaque.

So, there are two sorts of problems to be addressed in developing a logical formalism for intentional notions:
- a syntactic one (intentional notions refer to sentences); and
- a semantic one (no substitution of equivalents).

Thus any formalism can be characterised in terms of two attributes: its language of formulation, and semantic model [p83].
Formalising Attitudes III

- Two fundamental approaches to the syntactic problem:
  - use a modal language, which contains modal operators, which are applied to formulae;
  - use a meta-language: a first-order language containing terms that denote formulae of some other object-language.
- We will focus on modal languages, and in particular, normal modal logics, with possible worlds semantics.

Possible-Worlds Semantics for Modal Logics

- The possible-worlds model for epistemic logics was proposed by Hintikka in 1962.
- It is now commonly formulated in a normal modal logic using techniques developed by Kripke in 1963.
- Hintikka’s insight: an agent’s belief in terms of a set of possible worlds.
- Example: agent playing a card game.
- Two advantages:
  - remains neutral on the subject of the cognitive structure of agents
  - appealing mathematical theory

General Normal Modal Logics

- Designed by philosophers interested in the distinction between necessary truths and contingent truths.
- Syntax is classical propositional logic extended by the addition of two operators ‘□’ (necessarily) and ‘♦’ (possibly).
- Let $\Phi = \{p, q, r, \ldots\}$ be a countable set of atomic propositions
  - If $p \in \Phi$, then $p$ is a formula.
  - If $\varphi, \psi$ are formulae, then so are $true, \neg \varphi, \varphi \lor \psi$.
  - If $\varphi$ is a formulae, then so are $\Box \varphi, \Diamond \varphi$. 

Normal Modal Logic Semantics

- Normal modal logics are concerned with truth at world; models therefore contain
  - a set of worlds \( W \);
  - a binary relation, \( R \), on \( W \), saying which worlds are possible relative to other worlds;
  - a valuation function \( \pi \), saying what propositions are true at each world
- For example, if \((w, w') \in R\), then if the agent was actually in world \( w \), then as far as it was concerned, he might be in world \( w' \)

Normal Modal Logic Semantics II

- A model for a normal propositional modal logic is a triple \( \langle W, R, \varphi \rangle \).
- The semantics of the language is given via the satisfactory relation \( \models \) which holds between pairs of the form \( \langle M, w \rangle \) (where \( M \) is a model, and \( w \) a reference world), and formulae of the language

\[
\langle M, w \rangle \models p \quad \text{where} \quad p \in \Phi, \text{iff} p \in \pi(w)
\]

\[
\langle M, w \rangle \models \neg \varphi \quad \text{iff} \langle M, w \rangle \not\models \varphi
\]

\[
\langle M, w \rangle \models \varphi \land \psi \quad \text{iff} \langle M, w \rangle \models \varphi \text{ or } \langle M, w \rangle \models \psi
\]

\[
\langle M, w \rangle \models \square \varphi \quad \text{iff} \forall w' \in W : (w, w') \in R \Rightarrow \langle M, w' \rangle \models \varphi
\]

\[
\langle M, w \rangle \models \lozenge \varphi \quad \exists w' \in W : (w, w') \in R \Rightarrow \langle M, w' \rangle \models \varphi
\]
Validity and Satisfiability

A formula is:
- **satisfiable** if it is satisfied for some model/world pair
- **unsatisfiable** if it is not satisfied by any model/world pair
- **true** in a model if it is satisfied for every world in the model
- **valid** in a class of models if it is true in every model in the class
- **valid** if it is true in the class of all models

If $\varphi$ is valid, we indicate this by writing $\models \varphi$. This notion is similar to the notion of ‘tautology’ in classical propositional logic - all tautologies are valid.

The Basic Modal Properties

1. $\models (\varphi \rightarrow \psi) \rightarrow (\square \varphi \rightarrow \square \psi)$. This axiom is called $K$, in honour of Kripke. It will be a theorem of any complete axiomatisation of normal modal logic.

2. $\models \varphi \rightarrow \square \varphi$. This property is called the necessitation rule. It will appear as a rule of inference in any axiomatisation of normal modal logic.

This two properties turn out to be the most problematic features of normal modal logics when they are used as logics of knowledge/belief. This exactly what we do for agents.

Correspondence theory

<table>
<thead>
<tr>
<th>Name</th>
<th>Axiom</th>
<th>Condition on $R$</th>
<th>First-Order characterisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>$\square \varphi \rightarrow \varphi$</td>
<td>Reflexive</td>
<td>$\forall w \in W \cdot (w, w) \in R$</td>
</tr>
<tr>
<td>D</td>
<td>$\square \varphi \rightarrow \Box \varphi$</td>
<td>Serial</td>
<td>$\forall w \in W \cdot \exists w' \in W \cdot (w, w') \in R$</td>
</tr>
<tr>
<td>4</td>
<td>$\square \varphi \rightarrow \square \Box \varphi$</td>
<td>Transitively</td>
<td>$\forall w, w', w'' \in W \cdot (w, w') \in R \land (w', w'') \in R \rightarrow (w, w'') \in R$</td>
</tr>
<tr>
<td>5</td>
<td>$\Box \varphi \rightarrow \Box \Box \varphi$</td>
<td>Euclidean</td>
<td>$\forall w, w', w'' \in W \cdot (w, w') \in R \land (w', w'') \in R \rightarrow (w, w'') \in R$</td>
</tr>
</tbody>
</table>
Modal Logic Systems

- A system of logic can be thought of as a set of formulae valid in some class of models.
- A member of the set is called a theorem of the logic (denoted as $\vdash \varphi$ for a theorem $\varphi$).
- KT is known as T, KT4 is known as S4, KD45 is known as weak-S5, KT5 is known as S5.

Modal Logics as Epistemic Logic I

- In epistemic logic, the formula $\Box \varphi$ is read as “it is known that $\varphi$.”
- Worlds are epistemic alternatives, accessibility relation defines what the alternatives are from a given world.
- To deal with multi-agent knowledge, one adds to the model structure an indexed set of accessibility relations, one for each agent.
- A model is then: $\langle W, R_1, \ldots, R_n, \pi \rangle$

Modal Logics as Epistemic Logic II

- The modal operator $\Box$ is replaced by an indexed set of unary modal operators $\{K_i\}$.
- $K_i \varphi$ is read as “$i$ knows that $\varphi$.”
- The semantics rule for $\Box$ is replaced by:
  $$\langle M, w \rangle \models K_i \varphi \iff \forall w' \in W \cdot (w, w') \in R_i \Rightarrow \langle M, w' \rangle \models \varphi$$
**Epistemic Logic and the Basic Properties**

- The necessitation rule:
  - Tells us that an agents knows all valid formulae
  - Amongst other things, knows all tautologies
  - Infinite many
  - Counterintuitive property

- Axiom K
  - Tells us that an agent's knowledge is closed under implication
  - Implies that an agent's knowledge is closed under logical consequence
  - Seems counterintuitive

- Combination is an overstrong property

---

**Logical Omniscience**

- Those two problems - that knowing all valid formulae, and that of knowledge/belief being closed under logical consequence - together constitute the famous **logical omniscience** problem.

- Has some damaging properties
  - Consistency $\varphi \vDash \varphi$, too strong properties
  - Equivalent propositions are not equivalent as beliefs $\varphi \vDash \psi$, propositions are too coarse grained for beliefs

---

**Axioms of Knowledge and Belief**

- Axiom T is the knowledge axiom: it says that what is known is true.
- Axiom D is the consistency axiom: if you know $\varphi$, you can’t also know $\neg \varphi$.
- Axiom 4 is positive introspection: if you know $\varphi$, you know you know $\varphi$.
- Axiom 5 is negative introspection: you are aware of what you don’t know.
- We can (to a certain extent) pick and choose which axioms we want to represent our agents.
- Often S5 is chosen as a logic of idealised knowledge.
- Weak-S5 is often chosen as a logic of idealised belief.
Common Knowledge I

- Often useful to be able to reason about “cultural” knowledge: things that everyone knows, and that everyone knows that everyone knows ...
- This is called **common knowledge**
- Example: Muddy children puzzle
- Extending the language with an extra modal operator $E$ which can be read as “everyone knows”.

$$E\phi \equiv K_1\phi \land ... \land K_n\phi \land K_{n+1}\phi$$

Common Knowledge II

- Derived operator $E^k$
- $E^k\phi$ can be read as “everyone knows $\phi$ to degree $k$
- It is defined as follows:
  $$E^{k+1}\phi \equiv E(E^k\phi)$$
- The Common Knowledge operator $C$ can then be defined as:
  $$C\phi \equiv E\phi \land E^2\phi \land ... \land E^k\phi \land ...$$
- Infinite conjunction
- The coordinated Attack problem

MetateM I

- MetateM is a framework for directly executing temporal logic specifications
- MetateM program is a collection of past $\Rightarrow$ future rules.
- Execution proceeds by a process of continually matching rules against a “history”, and firing those rules whose antecedents are satisfied.
- The instantiated future-time consequents become commitments which must subsequently be satisfied.
- Execution is thus a process of iteratively generating a model for the formulae made up by the program rules
- The future-time parts of instantiated rules represent constraints on this model
**Why Theory?**

- Modal Logic
- Epistemic Logic
- Other Logics with Modal Operators
- Time as a Modality
- BDI Logic

---

**MetateM II**

- An example MetateM program: the resource controller:

\[ \forall x \text{ask}(x) \Rightarrow \Diamond \text{give}(x) \]

\[ \forall x, y \text{give}(x) \land \text{give}(y) \Rightarrow (x = y) \]

- First rule ensures that an `ask` is eventually followed by a `give`.
- Second rule ensures that only one `give` is ever performed at any one time.

---

**Concurrent MetateM (I)**

- Concurrent MetateM provides an additional framework through which societies of MetateM processes can operate and communicate.
- It is based on a new model for concurrency in executable logics: the notion of executing a logical specification to generate individual agent behaviour.
- Currently used in a project for personalised software museum guides.
- A Concurrent MetateM system contains a number of agents (objects), each object has 3 attributes:
  - a name;
  - an interface;
  - a MetateM program.

---

**Concurrent MetateM (II)**

- An agent's interface contains two sets:
  - messages the agent will accept;
  - messages the agent may send.
- For example, a `stack` object's interface:

\[ \text{stack}(\text{pop}, \text{push})[\text{popped}, \text{stackfull}] \]

\[ \{\text{pop}, \text{push}\} = \text{messages received} \]

\[ \{\text{popped}, \text{stackfull}\} = \text{messages sent} \]

- If an object (agent) satisfies a commitment corresponding to a sendable message, it broadcasts it.
White and the Seven Dwarfs

- To illustrate the language Concurrent MetateM in more detail, here are some example programs.
- Snow White has some sweets (resources), which she will give to the Dwarfs (resource consumers).
- She will only give to one dwarf at a time.
- She will always eventually give to a dwarf that asks.
- Here is Snow White, written in Concurrent MetateM:
  
  SnowWhite(ask)[give]:
  ⊩ ask(x) ⇒ ⊤ give(x)
  give(x) ∧ give(y) ⇒ (x = y)

More Dwarfs

- The dwarf ‘eager’ asks for a sweet initially, and then whenever he has just received one, asks again.
  
  eager(give)[ask]:
  start ⇒ ask(eager)
  ⊩ give(eager) ⇒ ask(eager)

- Some dwarfs are even less polite: ‘greedy’ just asks every time.
  
  greedy(give)[ask]:
  start ⇒ □ ask(greedy)

And More Dwarfs

- Fortunately, some have better manners; ‘courteous’ only asks when ‘eager’ and ‘greedy’ have eaten.
  
  courteous(give)[ask]:
  ((¬ ask(courteous) S give(eager)) ∧
  (¬ ask(courteous) S give(greedy))) ⇒ ask(courteous)

- And finally, ‘shy’ will only ask for a sweet when noone else has just asked.
  
  shy(give)[ask]:
  start ⇒ ◊ ask(shy)
  ◩ ask(x) ⇒ ¬ ask(shy)
  ◩ give(shy) ⇒ ◊ ask(shy)
**BDI Theory and Practise**

- We consider the *semantics* of BDI architectures: to what extent does a BDI agent satisfy a theory of agency.
- In order to give a semantics to BDI architectures, Rao & Georgeff gave developed BDI logics: non-classical logics with modal connectives for representing beliefs, desires, and intentions.
- The 'basic BDI logic' of Rao & Georgeff is a quantified extension of the expressive branching time logic $CTL^*$. 

**BDI Logic**

- From classical logic: $\land, \lor, \neg, \ldots$
- The $CTL^*$ path quantifiers:
  
  - $A\phi$ ‘on all paths, $\phi$'
  - $E\phi$ ‘on some paths, $\phi$'
- The BDI connectives:
  
  - $(Bel i \phi)$ $i$ believes $\phi$
  - $(Des i \phi)$ $i$ desires $\phi$
  - $(Int i \phi)$ $i$ intends $\phi$
- Let us now look at some possible axioms of BDI logic, and see to what extent the BDI architecture could be said to satisfy these axioms.

**Axioms (I)**

- Belief goal compatibility:
  
  $$(Des \alpha) \Rightarrow (Bel \alpha)$$
  
  States that if the agent has a goal to optionally achieve something, this thing must be an option.
  
  This axiom is operationalised in the function $options$: an option should not be produced if it is not believed possible.
- Goal Intention compatibility:
  
  $$(Int \alpha) \Rightarrow (Des \alpha)$$
  
  States that having an intention to optionally achieve something implies having it as goal (i.e. there are no intentions that are not goals).
  
  Operationalised in the $deliberate$ function.
**Axioms (II)**

- **Volitional commitment**
  
  \[(\text{Int } \text{does}(a)) \Rightarrow \text{does}(a)\]

  If you intend to perform some action \(a\) next, then you \(a\) next.

  Operationalised in the \textit{execute} function.

- **Awareness of goals and intentions**
  
  \[
  (\text{Des } \phi) \Rightarrow (\text{Bel } (\text{Des } \phi)) \\
  (\text{Int } \phi) \Rightarrow (\text{Bel } (\text{Int } \phi))
  \]

  Requires that new intentions and goals be posted as events.

**Axioms (III)**

- **No unconscious actions**
  
  \[
  \text{done}(a) \Rightarrow (\text{Bel } \text{done}(a))
  \]

  If an agent does some action, then it is aware that it has done the action.

  Operationalised in the \textit{execute} function.

  A stronger requirement would be for the success or failure of the action posted.

- **No infinite deferral**
  
  \[
  (\text{Int } \phi) \Rightarrow A \diamond (\neg (\text{Int } \phi))
  \]

  An agent will eventually either act for an intention, or else drop it.

**Summary**

- **Modal Logic**
  
  - Extension from classical logic
  - Possible world view
  - Axiom K
  - Logical Omniscience

- **Epistemic Logic**
  
  - The multi agent view on modal logic
  - Common knowledge
Directed and Additional Reading

- Wooldridge Chapter 12.
- Wooldridge Chapter 3.
- Cohen and Levesque: Intention is Choice with Commitment
