An illustration

1. The agent (A) receives an "utterance" from another agent (B), which we assume is a sentence in the form of a sequence of words.
2. The agent (A) "parses" the sentence.
3. The agent assigns "meaning" to the sentence, perhaps in the form of a first order logic statement about a model of the world that the agent constructs.
4. The agent forms a plan to react to the utterance. To carry out the plan the agent A needs to enlist the cooperation of agent B (or perhaps some other agent). This cooperation will take the form of a sequence of actions by agent B.
5. Agent A asks B to perform the action. This requires (sometimes) that A compose an utterance to deliver to B.
Components Needed
to make the model work.
(Using a Physical Symbol system
PSS model)

• A Parser/Interpreter that turns the input sentence into a statement/question about the agent's internal model.
• An internal model of the world.
• A module that can reason about the internal model.
• A planning module that determines what action to take.
• A module that takes the requisite action.
Suppose that we want to build a system that understands sentences such as

Is block 3 red?
Which blocks are green?
Put block 3 on block 4?
Where is block 5?

We can give it a model of the world that contains a collection of assertions:

red(block(3)).
on(block(3), block(2)).
on(block(2), table).
green(block(2)).
We give it a system that connects input sentences to their "meaning".

% Unprincipled grammar.

sentence(true) --> [is],
  noun_phrase(sing, SemN),
  adj(SemAdj),
  {G =..[SemAdj, SemN], call(G)}.

sentence(Semantics) --> [put],
  noun_phrase(sing, SemN1),
  p_phrase(SemP, SemN2),
  {Semantics =
    a_plan_to( put_on( SemN1, SemN2))}.

sentence(Semantics) --> [where], [is],
  noun_phrase(sing, SemN),
  { on(SemN, Semantics)}.

sentence(Semantics) --> [which],
  noun_phrase(plur, SemN), [are],
  adjective(SemAdj),
  { G =..[SemAdj, SemN],
    setof(SemN, call(G), Semantics)}. 
noun_phrase(sing, block(N)) --> [block], num(N).
noun_phrase(plur, block(X)) --> [blocks].
num(N) --> [N], {number(N)}.

p_phrase(on, block(N)) --> [on, block], num(N).
p_phrase(on, table) --> [on, the, table].
adjective(red) --> [red].
adjective(green) --> [green].

% The World
red(block(3)).
on(block(3), block(2)).
on(block(2), table).
green(block(2)).
This model has the reasoner (for true) built in from the Prolog engine.

You can see where the planner would come in.

At a certain level language generation is just the parser used backwards.
Fundamental AI Concepts

- Knowledge Representation (the world)
- Reasoning.
- Planning,
- Action
- Sensing.

The key to understanding AI lies in realizing that it deals with ill posed problems where scale is critical. In general the goal is to find solutions that in Alan Newell's word "satisfice".

test(S):- sentence(Sem, S, []), write(Sem).