# Design for an OpenCog Dialogue System Inspired by Speech Act Theory

Ben Goertzel and Ruiting Lian

March 21, 2011

**Abstract**

A design for a dialogue system inspired by a variant of Speech Act Theory is roughly sketched, intended for implementation within the OpenCog integrative AGI system. The goal is not to enable precisely human-like dialogue, but rather to enable cognitively general, emotionally expressive, pragmatically appropriate and experientially grounded dialogue. The main target applications are dialogue for game characters and humanoid robots, but the same ideas could also be applied in the context of purely text-based dialogue systems, e.g. conversational search interfaces. A multi-phase approach to implementation is outlined, beginning with a relatively small and simple system, and ending with a system potentially capable of approaching human-level functionality.

## 1 Introduction

*Reader beware: this is a preliminary document, intended to guide ongoing development. It lacks references and may be missing important details. It will be fleshed out as development proceeds.*

Most natural language dialogue systems in practical use today are based in some way on relatively simplistic sets of "template rules". The systems best at emulating human conversation, according to tests like the Loebner Prize competition, are hybrids of ELIZA-style rule-based chat engines with statistical learning system (that have learned conversational patterns via statistically analyzing corpuses of conversations). Other, purpose-specific dialogue systems operate based on sets of rules particular to a certain narrow application domain (say, finding the user information about a certain topic like the weather, or finding information within a certain document repository, or allowing the user to control a certain machine). These systems may utilize nontrivial natural language processing approaches for both comprehension and generation, but the "purposeful cognition" aspect intervening between comprehension and generation is carried out via specialized and often rather brittle and/or domain-specific rules.

Of course, based on what we currently believe about psycholinguistics and neurolinguistics, a truly humanistic dialogue system would operate quite differently. A young child learns language in a manner that's intimately bound up with sound and gesture, and with the process of learning to perceive and act and socialize and generally interpret the world. Syntax, semantics, pragmatics and phonology are acquired in an integrated way. Experiential learning is guided by neurally-coded inductive biases that infuse into the mind progressively, partially triggered, at each stage, by the linguistic competencies already acquired.

Creating a "humanistic" AI dialogue system of this nature would be a wonderful challenge, but seems an extraordinarily difficult research project given the current states of the various supporting technologies and sciences. What we suggest here is a sort of middle ground between the state of the art, comprising systems governed by rules and corpus statistics, and the humanistic approach. An intermediate system of this nature may be viewed as a partial step toward a more humanistic system, or as a practical way to get additional functionality into a dialogue system without going all the way to a humanistic approach.

To understand the particulars of this document, you will need to understand:

- the basics of the current OpenCog NLP framework, e.g. the link parser, RelEx, RelEx2Frame and NLGen.

- the basics of the OpenCog architecture (Atomspace, MindAgents, etc.) as well as OpenPsi

The dialogue system we are considering has two phases of development:

1. **Phase 1**:
   - ”Lower levels” of NL comprehension and generation executed by a relatively traditional approach incorporating statistical and rule-based aspects (the RelEx and NLGen systems)
   - Dialogue control utilizes hand-coded procedures and predicates (SpeechActSchema and SpeechActTriggers) corresponding to fine-grained types of speech act
   - Dialogue control guided by general cognitive control system (OpenPsi, running within OpenCog)
   - SpeechActSchema and SpeechActTriggers, in some cases, will internally consult probabilistic inference, thus supplying a high degree of adaptive intelligence to the conversation
2. **Phase 2**:
   - ”Lower levels” of NL comprehension and generation carried out within primary cognition engine, in a manner enabling their underlying rules and probabilities to be modified based the system's experience. Concretely, one way this could be done in OpenCog would be via
     - Implementing the RelEx and RelEx2Frame rules as PLN implications in the Atomspace
     - Implementing parsing via expressing the link parser dictionary as Atoms in the Atomspace, and using the SAT link parser to do parsing as an example of logical unification (carried out by a MindAgent wrapping an SAT solver)
     - Implementing NLGen within the OpenCog core, via making NLGen's sentence database a specially indexed Atomspace, and wrapping the NLGen operations in a MindAgent
   - Reimplement the SpeechActSchema and SpeechActTriggers in an appropriate combination of Combo and PLN logical link types, so they are susceptible to modification via inference and evolution

In this brief document we will focus mainly on Phase 1, but we mention Phase 2 so that the overall direction of intended development is clear. It's worth noting that the work required to move from Phase 1 to Phase 2 is essentially software development and computer science algorithm optimization work, rather than computational linguistics work per se. Then after the Phase 2 system is built there will, of course, be significant work involved in enabling PLN, MOSES and other cognitive algorithms to experientially adapt the various portions of the dialogue system that have been moved into the OpenCog core and refactored for adaptiveness.

## 2 Speech Act Theory and its Elaboration

We review here the very basics of speech act theory, and then the specific variant of speech act theory that we feel will be most useful for practical OpenCog dialogue system development.

The core notion of speech act theory is to analyze linguistic behavior in terms of discrete speech acts aimed at achieving specific goals. This is a convenient theoretical approach in an OpenCog context, because it pushes us to treat speech acts just like any other acts that an OpenCog system may carry out in its world, and to handle speech acts via the standard OpenCog action selection mechanism.

Searle, who originated speech act theory, divided speech acts according to the following (by now well known) ontology:

- **Assertives** : The speaker commits herself to something being true. *The sky is blue.*
- **Directives**: The speaker attempts to get the hearer to do something. *Clean your room!*
- **Commissives**: The speaker commits to some future course of action. *I will do it.*
- **Expressives**: The speaker expresses some psychological state. *Im sorry.*
- **Declarations**: The speaker brings about a different state of the world. *The meeting is adjourned.*

Inspired by this ontology, Twitchell and Nunamaker (in their 2004 paper "Speech Act Profiling: A Probabilistic Method for Analyzing Persistent Conversations and Their Participants") created a much more fine-grained ontology of 42 kinds of speech acts, called SWBD-DAMSL (DAMSL = Dialogue Act Markup in Several Layers). Nearly all of their 42 speech act types can be neatly mapped into one of Searle's 5 high level categories, although a handful don't fit Searle's view and get categorized as "other." Figures **??** and **??** depict the 42 acts and their relationship to Searle's categories.

# 3 Speech Act Schemata and Triggers

In the suggested dialogue system design, multiple SpeechActSchema would be implemented, corresponding *roughly* to the 42 SWBD-DAMSL speech acts. The correspondence is "rough" because

- we may wish to add new speech acts not in their list
- sometimes it may be most convenient to merge 2 or more of their speech acts into a single SpeechActSchema. For instance, it's probably easiest to merge their YES ANSWER and NO ANSWER categories into a single TRUTH VALUE ANSWER schema, yielding affirmative, negative, and intermediate answers like "probably", "probably not", "I'm not sure", etc.
- sometimes it may be best to split one of their speech acts into several, e.g. to separately consider STATEMENTs which are responses to statements, versus statements that are unsolicited disbursements of "what's on the agent's mind."

Overall, the SWBD-DAMSL categories should be taken as guidance rather than doctrine. However, they are valuable guidance due to their roots in detailed analysis of real human conversations, and their role as a bridge between concrete conversational analysis and the abstractions of speech act theory.

Each SpeechActSchema would take in an input consisting of a DialogueNode, a Node type possessing a collection of links to

- a series of past statements by the agent and other conversation participants, with
  - each statement labeled according to the utterer
  - each statement uttered by the agent, labeled according to which SpeechActSchema was used to produce it, plus (see below) which SpeechActTrigger and which response generator was involved
- a set of Atoms comprising the context of the dialogue. These Atoms may optionally be linked to some of the Atoms representing some of the past statements. If they are not so linked, they are considered as general context.

The enaction of SpeechActSchema would be carried out via PredictiveImplicationLinks embodying "Context AND Schema → Goal" schematic implications, of the general form

```
PredictiveImplication
   AND
        Evaluation
           SpeechActTrigger T
           DialogueNode D
        Execution
          SpeechActSchema S
          DialogueNode D
   Evaluation
        Evaluation
           Goal G
```

with

```
ExecutionOutput
     SpeechActSchema S
     DialogueNode D
     UtteranceNode U
```

being created as a result of the enaction of the SpeechActSchema. (An UtteranceNode is a series of one or more SentenceNodes.)

A single SpeechActSchema may be involved in many such implications, with different probabilistic weights, if it naturally has many different Trigger contexts.

Internally each SpeechActSchema would contain a set of one or more response generators, each one of which is capable of independently producing a response based on the given input. These may also be weighted, where the weight determines the probability of a given response generation process being chosen in preference to the others, once the choice to enact that particular SpeechActSchema has already been made.

## 3.1 Notes Toward Example SpeechActSchema

To make the above ideas more concrete, let's consider a few specific SpeechActSchema. We won't fully specify them here, but will outline them sufficiently to make the ideas clear.

### 3.1.1 TruthValueAnswer

The TruthValueAnswer SpeechActSchema would encompass SWBD-DAMSL's YES ANSWER and NO ANSWER, and also more flexible truth value based responses.

**Trigger context** : when the conversation partner produces an utterance that RelEx maps into a truth-value query (this is simple as truth-value-query is one of RelEx's relationship types).

**Goal** : the simplest goal relevant here is pleasing the conversation partner, since the agent may have noticed in the past that other agents are pleased when their questions are answers. (More advanced agents may of course have other goals for answering questions, e.g. providing the other agent with information that will let it be more useful in future.)

**Response generation schema** : for starters, this SpeechActSchema could simply operate as follows. It takes the relationship (Atom) corresponding to the query, and uses it to launch a query to the pattern matcher or PLN backward chainer. Then based on the result, it produces a relationship (Atom) embodying the answer to the query, or else updates the truth value of the existing relationship corresponding to the answer to the query. This "answer" relationship has a certain truth value. The schema could then contain a set of rules mapping the truth values into responses, with a list of possible responses for each truth value range. For example a very high strength and high confidence truth value would be mapped into a set of responses like {definitely, certainly, surely, yes, indeed}.

This simple case exemplifies the overall Phase 1 approach suggested here. The conversation will be guided by fairly simple heuristic rules, but with linguistic sophistication in the comprehension and generation aspects, and potentially subtle inference invoked within the SpeechActSchema or (less frequently) the Trigger contexts. Then in Phase 2 these simple heuristic rules will be refactored in a manner rendering them susceptible to experiential adaptation.

### 3.1.2 Statement: Answer

The next few SpeechActSchema (plus maybe some similar ones not given here) are intended to collectively cover the ground of SWBD-DAMSL's STATEMENT OPINION and STATEMENT NON-OPINION acts.

**Trigger context** : The trigger is that the conversation partner asks a wh- question

**Goal** : Similar to the case of a TruthValueAnswer, discussed above

**Response generation schema** : When a wh- question is received, one reasonable response is to produce a statement comprising an answer. The question Atom is posed to the pattern matcher or PLN, which responds with an Atom-set comprising a putative answer. The answer Atoms are then pared down into a series of sentence-sized Atom-sets, which are articulated as sentences by NLGen. If the answer Atoms have very low-confidence truth values, or if the Atomspace contains knowledge that other agents significantly disagree with the agent's truth value assessments, then the answer Atom-set may have Atoms corresponding to "I think" or "In my opinion" etc. added onto it (this gives an instance of the STATEMENT NON-OPINION act).

#### 3.1.3 Statement: Unsolicited Observation

**Trigger context** : when in the presence of another intelligent agent (human or AI) and nothing has been said for a while, there is a certain probability of choosing to make a "random" statement.

**Goal 1** : Unsolicited observations may be made with a goal of pleasing the other agent, as it may have been observed in the past that other agents are happier when spoken to

**Goal 2** : Unsolicited observations may be made with goals of increasing the agent's own pleasure or novelty or knowledge – because it may have been observed that speaking often triggers conversations, and conversations are often more pleasurable or novel or educational than silence

**Response generation schema** : One option is a statement describing something in the mutual environment, another option is a statement derived from high-STI Atoms in the agent's Atomspace. The particulars are similar to the "Statement: Answer" case.

#### 3.1.4 Statement: External Change Notification

**Trigger context** : when in a situation with another intelligent agent, and something significant changes in the mutually perceived situation, a statement describing it may be made.

**Goal 1** : External change notification utterances may be made for the same reasons as Unsolicited Observations, described above.

**Goal 2** : The agent may think a certain external change is important to the other agent it is talking to, for some particular reason. For instance, if the agent sees a dog steal Bob's property, it may wish to tell Bob about this.

**Goal 3** : The change may be important to the agent itself – and it may want its conversation partner to do something relevant to an observed external change ... so it may bring the change to the partner's attention for this reason. For instance, "Our friends are leaving. Please try to make them come back."

**Response generation schema** : The Atom-set for expression characterizes the change observed. The particulars are similar to the "Statement: Answer" case.

#### 3.1.5 Statement: Internal Change Notification

**Trigger context 1** : when the importance level of an Atom increases dramatically while in the presence of another intelligent agent, a statement expressing this Atom (and some of its currently relevant surrounding Atoms) may be made

**Trigger context 2** : when the truth value of a reasonably important Atom changes dramatically while in the presence of another intelligent agent, a statement expressing this Atom and its truth value may be made

**Goal** : Similar goals apply here as to External Change Notification, considered above

**Response generation schema** : Similar to the "Statement: External Change Notification" case.

### 3.1.6 WHQuestion

**Trigger context** : being in the presence of an intelligent agent thought capable of answering questions

**Goal 1** : the general goal of increasing the agent's total knowledge

**Goal 2** : the agent notes that, to achieve one of its currently important goals, it would be useful to possess a Atom fulfilling a certain specification

**Response generation schema** : Formulate a query whose answer would be an Atom fulfilling that specification, and then articulate this logical query as an English question using NLGen

# 4 Probabilistic Mining of Trigger contexts

One question raised by the above design sketch is where the Trigger contexts come from. They may be hand-coded, but this approach may suffer from excessive brittleness. The approach suggested by Twitchell and Nunamaker's work (which involved modeling human dialogues rather than automatically generating intelligent dialogues) is statistical. That is, they suggest marking up a corpus of human dialogues with tags corresponding to the 42 speech acts, and learning from this annotated corpus a set of Markov transition probabilities indicating which speech acts are most likely to follow which others. In their approach the transition probabilities refer only to series of speech acts.

In an OpenCog context one could utilize a more sophisticated training corpus in a more sophisticated way. For instance, suppose one wants to build a dialogue system for a game character conversing with human characters in a game world. Then one could conduct experiments in which one human controls a "human" game character, and another human puppeteers an "AI" game character. That is, the puppeteered character funnels its perceptions to the AI system, but has its actions and verbalizations controlled by the human puppeteer. Given the dialogue from this sort of session, one could then perform markup according to the 42 speech acts.

As a simple example, consider the following brief snippet of annotated conversation:

| **speaker** | **utterance** | **speech act type** |
|---|---|---|
| Ben | Go get me the ball | ad |
| AI | Where is it? | qw |
| Ben | Over there [points] | sd |
| AI | By the table? | qy |
| Ben | Yeah | ny |
| AI | Thanks | ft |
| AI | I'll get it now. | commits |

A DialogueNode object based on this snippet would contain the information in the table, plus some physical information about the situation, such as, in this case: predicates describing the relative locations of the two agents, the ball an the table (e.g. the two agents are very near each other, the ball and the table are very near each other, but these two groups of entities are only moderately near each other); and, predicates involving

Then, one could train a machine learning algorithm such as MOSES to predict the probability of speech act type $S_1$ occurring at a certain point in a dialogue history, based on the prior history of the dialogue. This prior history could include percepts and cognitions as well as utterances, since one has a record of the AI system's perceptions and cognitions in the course of the marked-up dialogue.

One question is whether to use the 42 SWBD-DAMSL speech acts for the creation of the annotated corpus, or whether instead to use the modified set of speech acts created in designing SpeechActSchema. Either way could work, but we are mildly biased toward the former, since this specific SWBD-DAMSL markup scheme has already proved its viability for marking up conversations. It seems unproblematic to map probabilities corresponding to these speech acts into probabilities corresponding to a slightly refined set of speech acts. Also, this way the corpus would be valuable independently of ongoing low-level changes in the collection of SpeechActSchema.

In addition to this sort of supervised training in advance, it will be important to enable the system to learn Trigger contexts online as a consequence of its life experience. This learning may take two forms:

1. Most simply, adjustment of the probabilities associated with the PredictiveImplicationLinks between SpeechActTriggers and SpeechActSchema

2. More sophisticatedly, learning of new SpeechActTrigger predicates, using an algorithms such as MOSES for predicate learning, based on mining the history of actual dialogues to estimate fitness

In both cases the basis for learning is information regarding the extent to which system goals were fulfilled by each past dialogue. PredictiveImplications that correspond to portions of successful dialogues will be have their truth values increased, and those corresponding to portions of unsuccessful dialogues will have their truth values decreased. Candidate SpeechActTriggers will be valued based on the observed historical success of the responses they would have generated based on historically perceived utterances; and (ultimately) more sophisticatedly, based on the estimated success of the responses they generate. Note that, while somewhat advanced, this kind of learning is much easier than th procedure learning required to learn new SpeechActSchema.

# 5 Conclusion

We have sketched a design for an OpenCog-based dialogue system, intermediate in sophistication and ”humanity” between current dialogue systems (which tend to be based on brittle rules or statistical learning) and advanced human-like language learning. We have divided the development system into two phases, the latter verging on human-level linguistic sophistication (though with significant differences from human psycholinguistics), and have focused mainly on the former here, after articulating a conceptually clear (though implementationally nontrivial) path from the former to the latter.

In order to implement Phase 1 of the suggested approach, several steps are required

1. Implementation of a DialogueNode object, and heuristics to assess what background knowledge to include in it

2. Integration of dialogue control in to OpenPsi

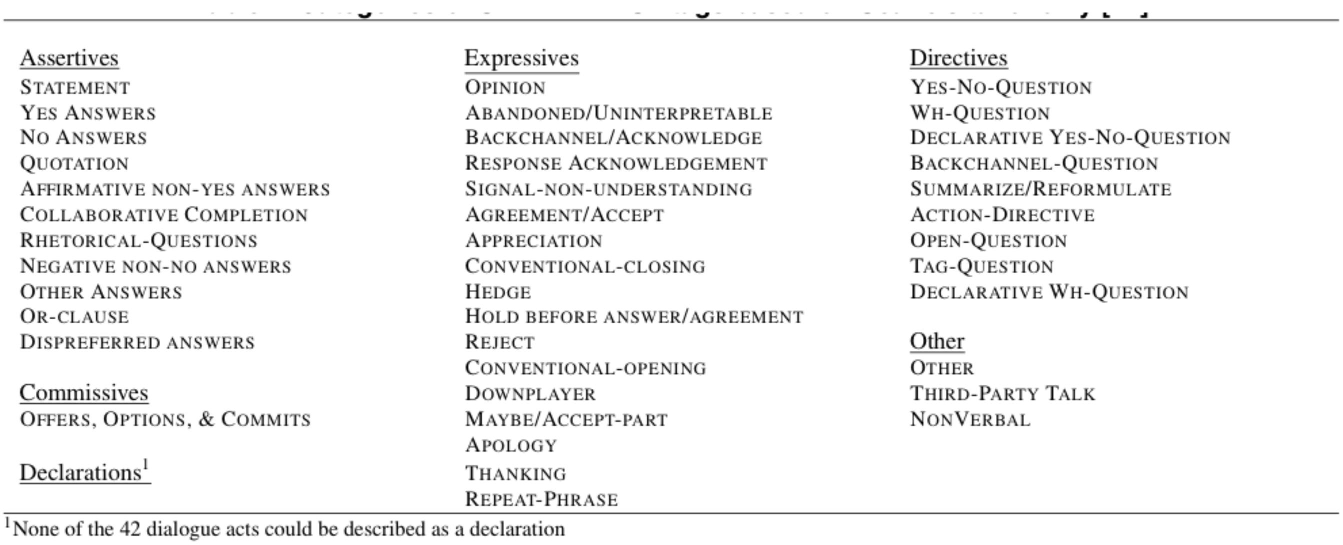
3. Implementation of SpeechActSchema corresponding *roughly* to the 42 SWBD-DAMSL speech acts

4. Creation of a marked-up corpus of ”puppeteered” embodied dialogues

While these steps are substantial, it’s worth noting that they can be executed partially, thus yielding a dialogue system with partial functionality. If simple but functional versions of items 1 and 2 are completed, then item 3 can be done for a limited number of speech acts, and hand-created SpeechActTriggers can initially be used in place of learned ones. Then more SpeechActSchema can be implemented gradually, and eventually a corpus can be created for inductive learning of triggers.

| Tag Name | Tag | Example |
|---|---|---|
| Statement-non-opinion | sd | Me, I'm in the legal department. |
| Acknowledge (Backchannel) | b | Uh-huh. |
| Statement-opinion | sv | I think it's great. |
| Agree/Accept | aa | That's exactly it. |
| Abandoned, Turn-Exit or Uninterpretable | % | So,- |
| Appreciation | ba | I can imagine. |
| Yes-No-Question | qy | Do you have to have any special training? |
| Non-verbal | x | [Laughter], [Throat-clearing] |
| Yes answers | ny | Yes. |
| Conventional-closing | fc | Well, it's been nice talking to you. |
| Wh-Question | qw | Well, how old are you? |
| No answers | nn | No. |
| Response Acknowledgement | bk | Oh, okay. |
| Hedge | h | I don't know if I'm making any sense or not. |
| Declarative Yes-No-Question | qyCd | So you can afford to get a house? |
| Other | other | Well give me a break, you know. |
| Backchannel in question form | bh | Is that right? |
| Quotation | Cq | You can't be pregnant and have cats. |
| Summarize/reformulate | bf | Oh, you mean you switched schools for the kids. |
| Affirmative non-yes answers | na | It is. |
| Action-directive | ad | Why don't you go first |
| Collaborative Completion | C2 | Who aren't contributing. |
| Repeat-phrase | bCm | Oh, fajitas |
| Open-Question | qo | How about you? |
| Rhetorical-Questions | qh | Who would steal a newspaper? |
| Hold before answer/agreement | Ch | I'm drawing a blank. |
| Reject | ar | Well, no |
| Negative non-no answers | ng | Uh, not a whole lot. |
| Signal-non-understanding | br | Excuse me? |
| Other answers | no | I don't know |
| Conventional-opening | fp | How are you? |
| Or-Clause | qrr | or is it more of a company? |
| Dispreferred answers | arp | Well, not so much that. |
| 3rd-party-talk | t3 | My goodness, Diane, get down from there. |
| Offers, Options Commits | commits | I'll have to check that out |
| Self-talk | t1 | What's the word I'm looking for |
| Downplayer | bd | That's all right. |
| Maybe/Accept-part | aap | Something like that |
| Tag-Question | Cg | Right? |
| Declarative Wh-Question | qwCd | You are what kind of buff? |
| Apology | fa | I'm sorry. |
| Thanking | ft | Hey thanks a lot |

Figure 1: The 42 DAMSL speech act categories.



Assertives

Statement
Yes Answers
No Answers
Quotation
Affirmative non-yes answers
Collaborative Completion
Rhetorical-Questions
Negative non-no answers
Other Answers
Or-clause
Dispreferred answers

Commissives

Offers, Options, & Commits

Declarations[1]

Expressives

Opinion
Abandoned/Uninterpretable
Backchannel/Acknowledge
Response Acknowledgement
Signal-non-understanding
Agreement/Accept
Appreciation
Conventional-closing
Hedge
Hold before answer/agreement
Reject
Conventional-opening
Downplayer
Maybe/Accept-part
Apology
Thanking
Repeat-Phrase

Directives

Yes-No-Question
Wh-Question
Declarative Yes-No-Question
Backchannel-Question
Summarize/Reformulate
Action-Directive
Open-Question
Tag-Question
Declarative Wh-Question

Other

Other
Third-Party Talk
NonVerbal

[1] None of the 42 dialogue acts could be described as a declaration

Figure 2: Connecting the 42 DAMSL speech act categories to Searle's 5 higher-level categories.