Discourse Pragmatics and Ellipsis Resolution in Task-Oriented Natural Language Interfaces

Jaime G. Carbonell

Computer Science Department
Carnegie-Mellon University,
Pittsburgh, PA 15213

Abstract

This paper reviews discourse phenomena that occur frequently in task-oriented man-machine dialogs, reporting on an empirical study that demonstrates the necessity of handling ellipsis, anaphora, extragrammaticality, inter-sentential metalanguage, and other abbreviatory devices in order to achieve convivial user interaction. Invariably, users prefer to generate terse or fragmentary utterances instead of longer, more complete "stand-alone" expressions, even when given clear instructions to the contrary. The XCALIBUR expert system interface is designed to meet these needs, including generalized ellipsis resolution by means of a rule-based caseframe method superior to previous semantic grammar approaches.

1. A Summary of Task-Oriented Discourse Phenomena

Natural language discourse exhibits several intriguing phenomena that defy definitive linguistic analysis and general computational solutions. However, some progress has been made in developing tractable computational solutions to simplified version of phenomena such as ellipsis and anaphora resolution [20, 10, 21]. This paper reviews discourse phenomena that arise in task-oriented dialogs with responsive agents (such as expert systems, rather than purely passive data base query systems), outlines the results of an empirical study, and presents our method for handling generalized ellipsis resolution in the XCALIBUR expert system interface. With the exception of inter-sentential metalanguage, and to a lesser degree extragrammaticality, the significance of the phenomena listed below have long been recognized and documented in the computational linguistics literature.

- **Anaphora** -- Interactive task-oriented dialogs invite the use of anaphora, much more so than simpler data base query situations.
- **Definite noun phrases** -- As Grosz [6] noted, resolving the referent of definite noun phrases requires an understanding of the planning structure underlying cooperative discourse.
- **Ellipsis** -- Sentential level ellipsis has long been recognized as ubiquitous in discourse. However, *semantic ellipsis*, where ellipsed information is manifest not as syntactically incomplete structures, but as semantically incomplete propositions, is also an important phenomenon. The ellipsis resolution method presented later in this paper addresses both kinds of ellipsis.
- **Extragrammatical utterances** -- Interjections, dropped articles, false starts, misspellings, and other forms of grammatical deviance abound in our data (as explained in the following section). Developing robust parsing techniques that tolerate errors has been the focus of our earlier investigations [2, 9, 7] and remains high among our priorities. Other investigations on error-tolerant parsing include [13, 22].
- **Meta-linguistic utterances** -- Intra-sentential metalanguage has been investigated to some degree [18, 12], but its more common inter-sentential counterpart has received little attention [4]. However, utterances about other utterances (e.g., corrections of previous commands, such as "I meant to type X instead" or "I should have said ...") are not infrequent in our dialogs, and we are making an initial stab at this problem [8]. Note that it is a cognitively less demanding task for a user to correct a previous utterance than to repeat an explicit sequence of commands (or worse yet, to detect and undo explicitly each and every unwanted consequence of a mistaken command).
- **Indirect speech acts** -- Occasionally users will resort to indirect speech acts [19, 16, 1], especially in connection with inter-sentential metalanguage or by stating a desired state of affairs and expecting the system to supply the sequence of actions necessary to achieve that state.

In our prior work we have focused on extragrammaticality and inter-sentential metalanguage. In this paper we report on an empirical study of discourse phenomena to a simulated interface and on our work on generalized ellipsis resolution in the context of the XCALIBUR project.

2. An Empirical Study

The necessity to handle most of the discourse phenomena listed in the preceding section was underscored by an empirical study we conducted to ascertain the most pressing needs of natural language interfaces in interactive applications. The initial objective of this study was to circumscribe the natural language interface task by attempting to instruct users of a simulated interface not to employ different discourse devices or difficult linguistic constructs. In essence, we wanted to determine whether untrained users would be able to interact as instructed (for instance avoiding all anaphoric referents), and, if so, whether they would still find the interface convivial given our artificial constraints.

The basic experimental set-up consisted of two remotely located terminals linked to each other and a transaction log file that kept a record of all interactions. The user was situated at one terminal and was told he or she was communicating with a real natural language interface to an operating system (and an accompanying intelligent help system, not unlike Wilensky's Unix Consultant [23].) The experimenter at the other terminal simulated the interface and gave appropriate commands to the (real) operating system.

In different sessions, users were instructed not to use pronouns, to type only complete sentences, to avoid complex syntax, to type only direct commands or queries (e.g., no indirect speech acts or discourse-level metalinguistic utterances [4, 8]), and to stick to the topic. The only instructions that were reliably followed were sticking to the topic (always) and avoiding complex syntax (usually). All other instructions were repeatedly violated in spite of constant negative feedback -- that is, the person pretending to be the natural language program replied with a standard error message. I recorded some verbal responses as well (with users telling a secretary at the terminal what she should type), and, contrary to my expectations, these did not qualitatively differ from the typed utterances. The significant result here is that users appear incapable or unwilling to generate lengthy commands, queries or statements when they can employ a linguistic device to state the same proposition in a more terse manner. To restate the principle more succinctly:

> **Terseness principle:** *users insist on being as terse as possible, independent of communication media or typing ability.*[1]

Given these results, we concluded that it was more appropriate to focus our investigations on handling abbreviatory discourse devices, rather than to address the issue of expanding our syntactic coverage to handle verbose complex structures seldom observed in our experience. In this manner, the objectives of the XCALIBUR project differ from those of most current investigations.

3. A Sketch of the XCALIBUR interface

This section outlines the XCALIBUR project, whose objective is to provide flexible natural language access (comprehension and generation) to the XSEL expert system [15]. XSEL, the Digital Equipment Corporation's automated salesman's assistant, advises on selection of appropriate VAX components and produces a sales order for automatic configuration by the R1 system [14]. Part of the XSEL task is to provide the user with information about DEC components, hence subsuming the database query task. However, unlike a pure data base query system, an expert system interface must also interpret commands, understand assertions of new information, and carry out task-oriented dialogs (such as those discussed by Grosz [6]). XCALIBUR, in particular, deals with commands to modify an order, as well as information requests pertaining to its present task or its data base of VAX component parts. In the near future it should process clarificational dialogs when the underlying expert system (i.e. XSEL) requires additional information or advice, as illustrated in the sample dialog below:

>*What is the largest 11780 fixed disk under $40,000?*

The rp07-aa is a 516 MB fixed pack disk that costs $38,000.

>*The largest under $50,000?*

The rp07-aa.

>*Add two rp07-aa disks to my order.*

Line item 1 added: (2 rp07-aa)

>*Add a printer with graphics capability*

fixed or changeable font?

>*fixed font*

lines per minute?

>*make it at least 200, upper/lowercase.*

Ok. Line item 2 added: (1 lxy11-sy)

>*Tell me about the lxy11*

The lxy11 is a 240 l/m line printer with plotting capabilities.

With the exception of the system-driven clarification interchange, which is beyond XCALIBUR's presently implemented capabilities, the rest of the dialog, including the natural language generation, is indicative of the present state of our system. The major contributions of XCALIBUR thus far is perhaps the integration of diverse techniques into a working system, including the DYPAR-II multi-strategy parser, expectation-based error correction, case-frame ellipsis resolution and focused natural language generation. Figure 3-1 provides a simplified view of the major modules of XCALIBUR, and the reader is referred to [3] for further elaboration.

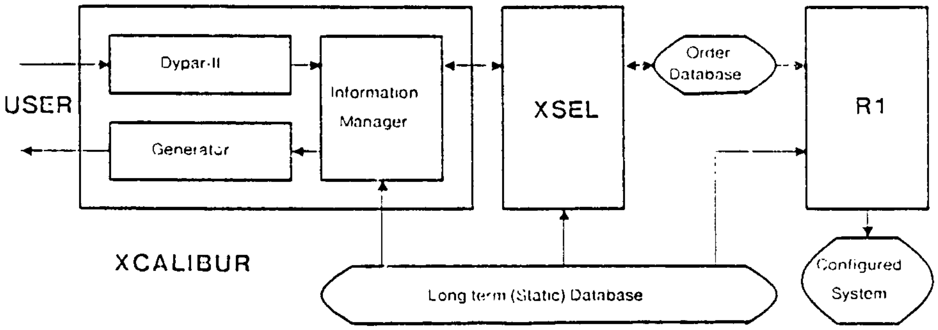

Figure 3-1: Overview of XCALIBUR

3.1. The Role of the Information Handler

When XSEL is ready to accept input, the information handler is

[1] Indicative as these empirical studies are of where one must focus one's efforts in developing convivial interfaces, they were not performed with adequate control groups or statistical rigor. Therefore, there is ample room to confirm, refute or expand upon the details of our empirical findings. However, the surprisingly strong form in which Grice's maxim [5] manifests itself in task-oriented human computer dialogs seems qualitatively irrefutable.

passed a message indicating the case frame or class of case frames expected as a response. For our example, assume that a command or query is expected, the parser is notified, and the user enters

>*What is the price of the 2 largest dual port fixed media disks?*

The parser returns:

```
[QUERY (OBJECT (SELECT (disk
                         (ports (VALUE (2)))
                         (disk-pack-type (VALUE (fixed)))
               (OPERATION (SORT
                            (TYPE (*descending))
                            (ATTR (size))
                            (NUMBER (2)))
               (PROJECT (price)) )
       (INFO-SOURCE (*default)) ]
```

Rather than delving into the details of the representation or the manner in which it is transformed prior to generating an internal command to XSEL, consider some of the functions of the information handler:

- Defaults must be instantiated. In the example, the query does not explicitly name an INFO-SOURCE, which could be the component database, the current set of line-items, or a set of disks brought into focus by the preceding dialog.
- Ambiguous fillers or attribute names must be resolved. For example, in most contexts, "300 MB disk" means a disk with "greater than or equal to 300 MB", rather than strictly "equal to 300 MB". A "large" disk refers to ample memory capacity in the context of a functional component specification, but to large physical dimensions during site planning. Presently, a small amount of local pragmatic knowledge suffices for the analysis, but, in the general case, closer integration with XSEL may be required.
- Generalized ellipsis resolution, as presented below, occurs within the information handler.

As the reader may note, the present *raison d'etre* of the information manager is to act as a repository of task and dialog knowledge providing information that the user did not feel necessary to convey explicitly. Additionally, the information handler routes the parsed command or query to the appropriate knowledge source, be it an external static data base, an expert system, or a dynamically constructed data structure (such as the current VAX order). Our plans call for incorporating a model of the user's task and knowledge state that should provide useful information to both parser and generator. At first, we intend to focus on stereotypical users such as a salesperson, a system engineer and a customer, who would have rather different domain knowledge, perhaps different vocabulary, and certainly different sets of tasks in mind. Eventually, refinements and updates to a default user model should be inferred from an analysis of the current dialog [17].

4. Generalized Caseframe Ellipsis

The XCALIBUR system handles ellipsis at the case-frame level. Its coverage appears to be a superset of the LIFER/LADDER system [10, 11] and the PLANES ellipsis module [21]. Although it handles most of the ellipsed utterances we encountered, it is not meant to be a general linguistic solution to the ellipsis phenomenon.

4.1. Examples

The following examples are illustrative of the kind of sentence fragments the current case-frame method handles. For brevity, assume that each sentence fragment occurs immediately following the initial query below.

INITIAL QUERY: "What is the price of the three largest
single port fixed media disks?"

"Speed?"

"Two smallest?"

"How about the price of the two smallest?"

"also the smallest with dual ports"

"Speed with two ports?"

"Disk with two ports."

In the representative examples above, punctuation is of no help, and pure syntax is of very limited utility. For instance, the last three phrases are syntactically similar (indeed, the last two are indistinguishable), but each requires that a different substitution be made on the preceding query. All three substitute the number of ports in the original SELECT field, but the first substitutes "ascending" for "descending" in the OPERATION field, the second substitutes "speed" for "price" in the PROJECT field, and the third merely repeats the case header of the SELECT field.

4.2. The Ellipsis Resolution Method

Ellipsis is resolved differently in the presence or absence of strong discourse expectations. In the former case, the discourse expectation rules are tested first, and, if they fail to resolve the sentence fragment, the contextual substitution rules are tried. If there are no strong discourse expectations, the contextual substitution rules are invoked directly.

Exemplary discourse expectation rule:

```
IF:   The system generated a query for confirmation or
      disconfirmation of a proposed value of a filler
      of a case in a case frame in focus.
THEN: EXPECT one or more of the following:
        1) A confirmation or disconfirmation pattern.
        2) A different but semantically permissible filler
           of the case frame in question (optionally naming
           the attribute or providing the case marker).
        3) A comparative or evaluative pattern.
        4) A query for possible fillers or constraints on
           possible fillers of the case in question.
           [If this expectation is confirmed, a sub-dialog
            is entered, where previously focused entities
            remain in focus.]
```

The following dialog fragment, presented without further commentary, illustrates how these expectations come into play in a focused dialog:

>*Add a line printer with graphics capabilities.*

Is 150 lines per minute acceptable?

>*No, 320 is better*	Expectations 1, 2 & 3
(or) *other options for the speed?*	Expectation 4
(or) *Too slow, try 300 or faster*	Expectations 2 & 3

The utterance "try 300 or faster" is syntactically a complete sentence, but semantically it is just as fragmentary as the previous utterances. The strong discourse expectations, however, suggest that it be processed in the same manner as syntactically incomplete utterances, since it satisfies the expectations of the interactive task. The terseness principle operates at all levels: syntactic, semantic and pragmatic.

The contextual substitution rules exploit the semantic representation of queries and commands discussed in the previous section. The scope of these rules, however, is limited to the last user interaction of appropriate type in the dialog focus, as illustrated in the following example:

Contextual Substitution Rule 1:

```
IF:   An attribute name (or conjoined list of attribute
      names) is present without any corresponding filler
      or case header, and the attribute is a semantically
      permissible descriptor of the case frame in the
      SELECT field or the last query in focus.
THEN: Substitute the new attribute name for the old filler
      of the PROJECT field of the last query.
```

For example, this rule resolves the ellipsis in the following utterances:

>What is the size of the 3 largest single port fixed media disks?
>And the price and speed?

Contextual Substitution Rule 2:

```
IF:   No sentential case frames are recognized in the
      input, and part of the input can be recognized as an
      attribute & filler (or just a filler) of a case in
      the SELECT field of a command or query in focus.
THEN: Substitute the new filler for the old in the same
      field of the old command or query.
```

This rule resolves the following kind of ellipsis:

>What is the size of the 3 largest single port fixed media disks?
>disks with two ports?

Note that it is impossible to resolve this kind of ellipsis in a general manner if the previous query is stored verbatim or as a a semantic-grammar parse tree. "Disks with two ports" would at best correspond to some <disk-descriptor> non-terminal, and hence, according to the LIFER algorithm [10, 11], would replace the entire phrase "single port fixed media disks" that corresponded to <disk-descriptor> in the parse of the original query. However, an informal poll of potential users suggests that the preferred interpretation of the ellipsis retains the MEDIA specifier of the original query. The ellipsis resolution process, therefore, requires a finer grain substation method than simply inserting the highest level non-terminals in the in the ellipsed input in place of the matching non-terminals in the parse tree of the previous utterance.

Taking advantage of the fact that a case frame analysis of a sentence or object description captures the meaningful semantic relations among its constituents in a canonical manner, a partially instantiated nominal case frame can be merged with the previous case frame as follows:

- Substitute any cases instantiated in the original query that the ellipsis specifically overrides. For instance "with two ports" overrides "single port" in our example, as both entail different values of the same case descriptor, regardless of their different syntactic roles. ("Single port" in the original query is an adjectival construction, whereas "with two ports" is a post-nominal modifier in the ellipsed fragment.)

- Retain any cases in the original parse that are not explicitly contradicted by new information in the ellipsed fragment. For instance, "fixed media" is retained as part of the disk description, as are all the sentential-level cases in the original query, such as the quantity specifier and the projection attribute of the query ("size").

- Add cases of a case frame in the query that are not instantiated therein, but are specified in the ellipsed fragment. For instance, the "fixed head" descriptor is added as the media case of the disk nominal case frame in resolving the ellipsed fragment in the following example:

>Which disks are configurable on a VAX 11-780?
>Any configurable fixed head disks?

- In the event that a new case frame is mentioned in the ellipsed fragment, wholesale substitution occurs, much like in the semantic grammar approach. For instance, if after the last example one were to ask "How about tape drives?", the substitution would replace "fixed head disks" with "tape drives", rather than replacing only "disks" and producing the phrase "fixed head tape drives", which is meaningless in the current domain. In these instances the semantic relations captured in a case frame representation and not in a semantic grammar parse tree prove immaterial.

The key to case-frame ellipsis resolution is matching corresponding cases, rather than surface strings, syntactic structures, or non-canonical representations. It is true that in order to instantiate correctly a sentential or nominal case frame in the parsing process requires semantic knowledge, some of which can be rather domain specific. But, once the parse is attained, the resulting canonical representation, encoding appropriate semantic relations, can and should be exploited to provide the system with additional functionality such as the present ellipsis resolution method.

The major problem with semantic grammars is that they convolve syntax with semantics in a manner that requires multiple representations for the same semantic entity. For instance, the ordering of marked cases in the input does not reflect any difference in meaning (although one could argue that surface ordering may reflect differential emphasis and other pragmatic considerations). A pure semantic grammar must employ different rules to recognize each and every admissible case sequence. Hence, the resultant parse trees differ, and the knowledge that surface positioning of unmarked cases is meaningful, but positioning of maked ones is not, must be contained within the ellipsis resolution process, a very unnatural repository for such basic information. Moreover, in order to attain a measure of the functionality described above for case-frames, ellipsis resolution in semantic grammar parse trees must somehow merge adjectival and post nominal forms (corresponding to different non-terminals and different relative positions in the parse trees) so that ellipsed structures such as "a disk with 1 port" can replace the the "dual-port" part of the phrase "...dual-port fixed-media disk ..." in an earlier utterance. One way to achieve this effect is to collect together specific nonterminals that can substitute for each other in certain contexts, in essence grouping non-canonical representations into semantic equivalence classes. However, this process would require hand-crafting large associative tables or similar data structures, a high price to pay for each domain-specific semantic grammar. Hence, in order to achive robust ellipsis resolution all proverbial roads lead to recursive case constructions encoding domain semantics and canonical structure for multiple surface manifestations.

Finally, consider one more rule that provides additional context in situations where the ellipsis is of a purely semantic nature, such as:

>*Which fixed media disks are configurable on a VAX780?*
The RP07-aa, the RP07-ab, ...
>*"Add the largest"*

We need to answer the question "largest what?" before proceeding. One can call this problem a special case of definite noun phrase resolution, rather than semantic ellipses, but terminology is immaterial. Such phrases occur with regularity in our corpus of examples and must be resolved by a fairly general process. The following rule answers the question from context, regardless of the syntactic completeness of the new utterance.

Contextual Substitution Rule 3:

```
IF:   A command or query caseframe lacks one or more
      required case fillers (such as a missing SELECT
      field), and the last case frame in focus has an
      instantiated case that meets all the semantic tests
      for the case missing the filler,
THEN: 1) Copy the filler onto the new caseframe, and
      2) Attempt to copy uninstantiated case fillers as
         well (if they meet semantic tests)
      3) Echo the action being performed for implicit
         confirmation by the user.
```

XCALIBUR presently has eight contextual substitution rules similar to the ones above, and we have found several additional ones to extend the coverage of ellipsed queries and commands (see [3] for a more extensive discussion). It is significant to note that a small set of fairly general rules exploiting the case frame structures cover most instances of commonly occurring ellipsis, including all the examples presented earlier in this section.

5. Acknowledgements

Mark Boggs, Peter Anick and Michael Mauldin are part of the XCALIBUR team and have participated in the design and implementation of various modules. Phil Hayes and Steve Minton have contributed useful ideas in several discussions. Digital Equipment Corporation is funding the XCALIBUR project, which provides a fertile test bed for our investigations.

6. References

1. Allen, J. F. and Perrault, C. R., "Analyzing Intention in Utterances," *Artificial Intelligence*, Vol. 15, No. 3, 1980, pp. 143-178.

2. Carbonell, J. G. and Hayes, P. J., "Dynamic Strategy Selection in Flexible Parsing," *Proceedings of the 19th Meeting of the Association for Computational Linguistics*, 1981.

3. Carbonell, J. G., Boggs, W. M., Mauldin, M. L. and Anick, P. G., "XCALIBUR Progress Report # 1: Overview of the Natural Language Interface," Tech. report, Carnegie-Mellon University, Computer Science Department, 1983.

4. Carbonell, J. G., "Beyond Speech Acts: Meta-Language Utterances, Social Roles, and Goal Hierarchies," *Preprints of the Workshop on Discourse Processes*, Marseilles, France, 1982.

5. Grice, H. P., "Conversational Postulates," in *Explorations in Cognition*, D. A. Norman and D. E. Rumelhart, eds., Freeman, San Francisco, 1975.

6. Grosz, B. J., *The Representation and Use of Focus in Dialogue Understanding*, PhD dissertation, University of California at Berkeley, 1977, SRI Tech. Note 151.

7. Hayes, P. J., and Carbonell, J. G., "Multi-Strategy Construction-Specific Parsing for Flexible Data Base Query and Update," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, August 1981, pp. 432-439.

8. Hayes, P. J. and Carbonell, J. G., "A Framework for Processing Corrections in Task-Oriented Dialogs," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983, (Submitted).

9. Hayes, P. J. and Carbonell, J. G., "Multi-Strategy Parsing and it Role in Robust Man-Machine Communication," Tech. report CMU-CS-81-118, Carnegie-Mellon University, Computer Science Department, May 1981.

10. Hendrix, G. G., Sacerdoti, E. D. and Slocum, J., "Developing a Natural Language Interface to Complex Data," SRI International, 1976.

11. Hendrix, G. G., "The LIFER Manual: A guide to Building Practical Natural Language Interfaces," Tech. report Tech. note 138, SRI, 1977.

12. Joshi, A. K., "Use (or Abuse) of Metalinguistic Devices", Unpublished Manuscript.

13. Kwasny, S. C. and Sondheimer, N. K., "Ungrammaticality and Extragrammaticality in Natural Language Understanding Systems," *Proceedings of the 17th Meeting of the Association for Computational Linguistics*, 1979, pp. 19-23.

14. McDermott, J., "R1: A Rule-Based Configurer of Computer Systems," Tech. report, Carnegie-Mellon University, Computer Science Department, 1980.

15. McDermott, J., "XSEL: A Computer Salesperson's Assistant," in *Machine Intelligence 10*, Hayes, J., Michie, D. and Pao, Y-H., eds., Chichester UK: Ellis Horwood Ltd., 1982", pp. 325-337.

16. Perrault, C. R., Allen, J. F. and Cohen, P. R., "Speech Acts as a Basis for Understanding Dialog Coherence," *Procceedings of the Second Conference on Theoretical Issues in Natural Language Processing*, 1978.

17. Rich, E., *Building and Exploring User Models*, PhD dissertation, Carnegie-Mellon University, April 1979.

18. Ross, J. R., "Metaanaphora," *Linguistic Inquiry*, 1970.

19. Searle, J. R., "Indirect Speech Acts," in *Syntax and Semantics, Volume 3: Speech Acts*, P. Cole and J. L. Morgan, eds., New York: Academic Press, 1975.

20. Sidner, C. L., *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*, PhD dissertation, MIT, 1979, AI-TR 537.

21. Waltz, D. L. and Goodman, A. B., "Writing a Natural Language Data Base System," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 1977, pp. 144-150.

22. Weischedel, R. M. and Black, J., "Responding to Potentially Unparsable Sentences," Tech. report, University of Delaware, Computer and Information Sciences, 1979, Tech Report 79/3.

23. Wilensky, R., "Talking to UNIX in English: An Overview of an Online Consultant," Tech. report, UC Berkeley, 1982.