A PRINCIPLED REPRESENTATION OF ATTRIBUTIVE DESCRIPTIONS FOR GENERATING INTEGRATED TEXT AND INFORMATION GRAPHICS PRESENTATIONS

Nancy Green*, Giuseppe Carenini, and Johanna Moore****
*Carnegie Mellon University, **University of Pittsburgh
nancy.green@cs.cmu.edu, {jmoore,carenini}@cs.pitt.edu

Abstract

This paper describes a media-independent, compositional, plan-based approach to representing attributive descriptions for use in integrated text and graphics generation. An *attributive* description's main function is to convey information directly contributing to the communicative goals of a discourse, whereas a *referential* description's *only* function is to enable the audience to identify a particular referent. This approach has been implemented as part of an architecture for generating integrated text and information graphics. Uses of referential and attributive descriptions are represented as two distinct types of communicative acts in a media-independent plan. It is particularly important to distinguish the two types of acts, since they have different consequences for dialogue and text generation, and for graphic design.

1 Introduction

This paper describes a media-independent, compositional, plan-based approach to representing attributive descriptions for use in integrated text and graphics generation. An *attributive* description's main function is to convey information directly contributing to the communicative goals of a discourse, whereas a *referential* description's *only* function is to enable the audience to identify a particular referent [Donnellan1977, Kronfeld1986]. While the generation of referential descriptions has received considerable attention in text and multimedia generation, the generation of attributive descriptions has received relatively little attention in computational linguistics.

However, such descriptions are pervasive in the type of presentations which is the focus of our research. We are developing systems that automatically generate presentations consisting of coordinated text and information graphics (graphics for presenting abstract, quantitative or relational information as opposed to depictions of real-world objects or processes). For example in our current implementation, the system produces analyses and summarizations of large amounts of data created by a transportation scheduling program. In this domain, it is necessary to generate descriptions of aggregate quantities of complex attributes such as *total port capacity of all ports* and *90% of the total weight of the cargo arriving by day 25.* Furthermore, in this genre both referential and attributive uses of descriptions occur.

In our approach, presentations are generated using an architecture that integrates hierarchical planning to achieve media-independent communicative goals with task-based graphic design. This architecture has been implemented in a prototype system. The focus of this paper is on the representation and role of attributive descriptions in the architecture. First, we describe the referential-attributive distinction and its importance in dialogue and text generation. Next, we discuss its importance in task-based graphic design. After providing an overview of our architecture, we describe how attributive descriptions are planned. We conclude with a survey of related work.

2 Referential-Attributive Distinction in Language

[Donnellan1977] describes two different possible uses of definite descriptions:[1] An *attributive* description's main function is to convey information directly contributing to the communicative goals of a discourse, whereas a *referential* description's *only* function is to enable the audience to identify a particular referent. This is a useful distinction for dialogue systems. In the case of failure of a referential description, a system might try to identify the referent again by giving an alternate description, as illustrated in (1) below. However, when a description is used attributively, the content of the description plays a different role. In (2a), *the required textbook for CS500* is used attributively to indirectly inform the user of how she might assess the difficulty of CS500 herself; the content of the description contributes to the user's recognition of the system's reason for suggesting that she read the book. In contrast in (2b), where an alternate description is used, the user is unable to recognize the systems's intention. In contrast to (1), (2c) illustrates that when an attributive description fails, a different type of followup by the system is required, one that explicates its intention. Also, as Donnellan points out and as can be seen by comparing (1) and (2c), the same description (*the required textbook for CS500*) can be used either referentially or attributively on different occasions depending on the speaker's intentions.

(1)	User:	I want to read an introductory AI book. What do you recommend?
	System:	The required textbook for CS500.
	User:	Which book is that?
	System:	Artificial Intelligence, by Richardson.
	User:	OK.
(2a)	User:	How difficult is CS500?
	System:	Read the required textbook for CS500.
(2b)	User:	How difficult is CS500?
	System:	Read Artificial Intelligence, by Richardson.
	User:	Why?
	System:	It is the required textbook for CS500.
(2c)	User:	How difficult is CS500?
	System:	Read the required textbook for it.
	User:	Why?
	System:	If you find the textbook difficult, then you'll find the course difficult.

In addition to its importance in determining appropriate dialogue followup behavior, the referential-attributive distinction is important for generating effective text. As was shown in (2a), the content of an attributive description may contribute directly to achieving communicative goals. To give another example, suppose that a user, who wants to buy a house in Somerset County, has asked for information about realtors serving Somerset County.[2] The overall goal of the system is for the user to believe that it may be beneficial to do business with a certain real estate agency, Realtors Inc. In that case, the system might generate (3), where (3)ii is intended to provide motivation for (3)i. That is, the description *the city with the largest population in Somerset County* was selected by the system for its motivational value. In a system that does not distinguish referential from attributive (i.e., treats all uses of descriptions as referential), there is nothing preventing it from generating (4) or (5) instead, assuming that *the city with the largest population in Somerset County*, *Berlin*, and *the city with the worst pollution in Somerset County* are three descriptions of the same object (which we refer to below by the internal system identifier $BERLIN).

[1] Although Donnellan did not address uses of indefinite descriptions, following [Kronfeld1990] we apply Donnellan's distinction to them as well. Also, to be precise, we are interested in what [Kronfeld1986, Kronfeld1990] terms the *modal aspect* of Donnellan's distinction.

[2] The information in this and all other examples in the paper is fictitious.

(3)i. We recommend Realtors Inc.
ii. Realtors Inc. serves the city with the largest population in Somerset County.
(4)i. We recommend Realtors Inc.
ii. Realtors Inc. serves Berlin.
(5)i. We recommend Realtors Inc.
ii. Realtors Inc. serves the city with the worst pollution in Somerset County.

However, (4)ii is not as effective as (3)ii if the user doesn't know or have in mind that Berlin has the largest population. Even worse, (5)ii might have an effect opposite to the one intended. A possible solution might be for the system to include as an additional proposition to be asserted with (4), the proposition that $BERLIN is the city with the largest population in Somerset County, yielding (6). On the other hand, there is nothing in the supposed underlyling representation of (6) to prevent (7) from being generated, which may have a less than desirable effect.

(6)i. We recommend Realtors Inc.
ii. Realtors Inc. serves Berlin.
iii. Berlin has the largest population in Somerset County.
(7)i. We recommend Realtors Inc.
ii. Realtors Inc. serves the city with the worst pollution in Somerset County.
iii. That city has the largest population in Somerset County.

3 The Role of Attributive Descriptions in Task-Based Graphic Design

As this section will illustrate shortly, different graphic designs may enhance or detract from a user's performance of certain types of perceptual and cognitive tasks. The philosophy of task-based graphic design is to design an information graphic based upon which perceptual and cognitive tasks the user wants or needs to perform. In our architecture (described more fully in the next section), the graphics generator reasons about what user tasks would enable the system's presentation goals to be achieved, so that graphics can be designed to support those tasks (and thus support the presentation goals). Note that since the descriptions in our domain of application are often fairly complex (e.g., *90% of the total weight of the cargo arriving by day 25*), we assume that a compositional approach to representing attributive descriptions will facilitate the automatic transformation of presentation goals to user tasks.

To see how different graphic designs about the same data may facilitate different tasks, consider Figure 1. In (a), the table shows that Arlington's population is .5K, Berlin's is 1K, etc. Moreover, it is possible to compute from the data shown in it that Arlington's population is half that of Berlin's, that Berlin has the largest population, and that Berlin's population is greater than the population of all of the other towns combined. To facilitate just task (A), the task of looking up the population of a town given its name, then this table would be adequate. On the other hand, a bar chart such as the one shown in (b) would better support both tasks (A) and (B), where (B) is the task of determining the largest and the smallest town. (Each vertical bar represents a particular town and the height of a bar represents the population of the town represented by the bar.) Ordering the towns by population size, as in (c), further facilitates task (B), as can be seen by comparing (b) to (c). However, task (C), the task of comparing Berlin's population to the total population of all of the other towns, would be facilitated by the chart shown in (d). In it, task (C) is facilitated by enabling the user to count the divisions of each bar. Also, if task (A) is not required, it is not necessary to provide numeric values on the horizontal axis in (d).

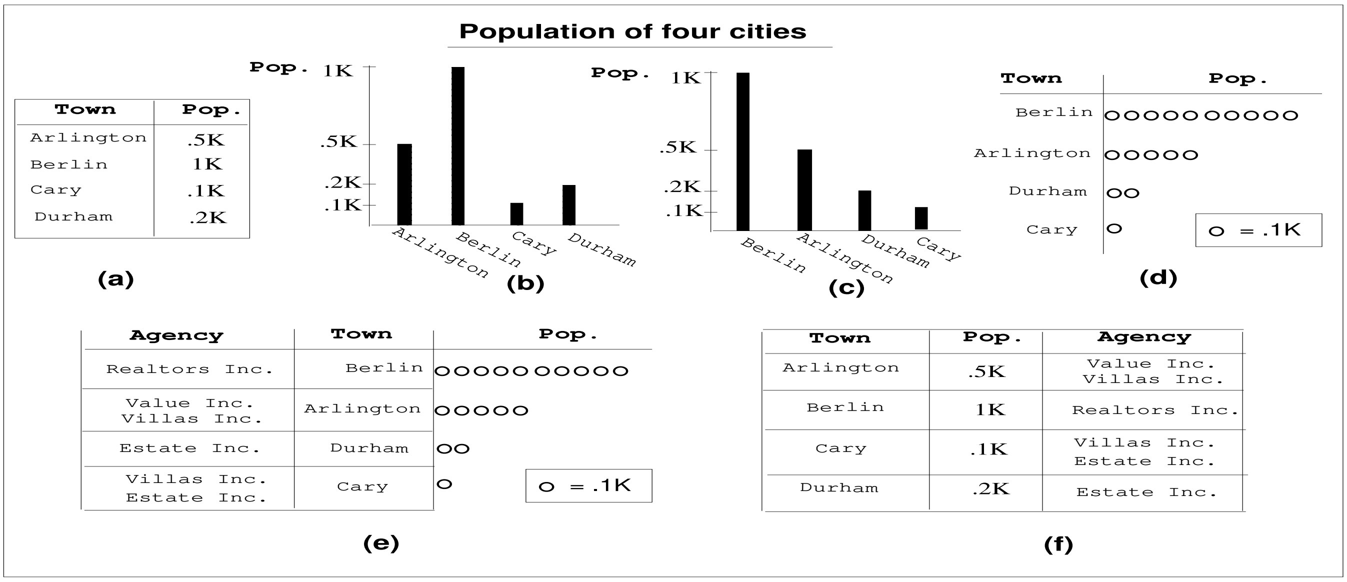

Figure 1: Graphics Supporting Different Tasks

Since in our approach the graphics generator reasons about what user tasks would enable the system's presentation goals to be achieved, it is important for the system to distinguish cases where the content of a description itself directly contributes to the presentation's goals, i.e., where the content has an attributive rather than a referential function. For instance, suppose that a system must design a graphic supporting the presentation goals described for example (3) above. These goals could be achieved by the user's successful performance of task (B) above, and additionally, task (D), the task of looking up the real estate agency for that town. These tasks would be facilitated by a graphic such as (e) in Figure 1, which facilitates both tasks. In contrast, if the system provided only table (f) of Figure 1, task (D) but not task (B) would be facilitated, and thus the overall presentation might not be as effective.

4 Overview of Generation Architecture

As reported in a previous paper [Kerpedjiev et al.1997], we are investigating the integration of two complementary approaches to automatic generation of presentations: hierarchical planning to achieve communicative goals and task-based graphic design. Many researchers in natural language processing, e.g., [Moore1995], have modeled presentation design as a process of hierarchical planning to achieve communicative goals. Researchers in graphics have emphasized the need to design presentations that support the perceptual and logical tasks a user must perform [Beshers and Feiner1993, Casner1991, Roth and Mattis1990]. In our hybrid approach, a hierarchical planner [Young1994] is used to refine genre-specific but media-independent presentation goals into genre-independent and media-independent subgoals. (For simplicity, in the rest of this paper we shall refer to the genre-independent and media-independent level of the plan just as the *media-independent* level.) These media-independent goals are achieved by media-independent illocutionary actions [Searle1970], e.g., *Assert*, and *Recommend*, which themselves are decomposed into media-independent actions that correspond to attributive and referential uses of descriptions. (The language used in our current system to express the *content* of illocutionary acts and goals is described in [Green et al.1998]. In addition to application-specific terms, the language includes more broadly applicable terms for expressing quantitative relations and aggregate properties.)

The media-independent plan is used by two media-specific generators (one for text, another for graphics) to create parts of the presentation. (The problems of *media-allocation*, how the system decides what parts of the presentation to realize in which media, and *media-coordination*, how it coordinates information conveyed in both media, are beyond the scope of this paper.) The text generator converts parts of the plan (as determined by the media-allocation component) to functional descriptions (FDs) of sentential units, which specify, for example, semantic predicate-argument structure, open-class lexical items, and aspects of sentence structure with pragmatic import. The FDs are subsequently realized by a general-purpose sentence generator (FUF/SURGE) [Elhadad and Robin1996]. (Decisions regarding the content of referential descriptions and anaphora, which are made by the text generator, are beyond the scope of this paper.) The first stage of the graphics generator converts parts of the plan (as determined by the media-allocation component) to a sequence of logical user tasks that will enable the presentation's goals to be achieved; the task sequence is then input to the SAGE graphic design system [Roth and Mattis1990, Chuah et al.1995, Roth et al.1994], which automatically creates a graphic supporting the user's tasks.[3] For example, the presentation goal that the user know the population of Arlington would be enabled if the user were able to perform the sequence of logical tasks of searching for Arlington in a graphic, finding its population attribute, and then looking up the value; furthermore, these tasks could be performed using a graphic such as (a) in Figure 1. (The process of converting acts of the plan to tasks is partly described in [Kerpedjiev et al.1998] and is beyond the scope of this paper.)

5 Planning Attributive Descriptions

This section describes how the two types of actions corresponding to attributive and referential uses of descriptions are created and represented in the media-independent planning phase of generation in our system. Our system uses media-independent presentation operators to perform content selection and high-level organization of the presentation. For example, Figure 2 shows a simplified version of the presentation operator that would be used to generate (3) above, in the formalism used by the presentation planner [Young1994]. The strategy encoded in this decomposition is to recommend an action, as in (3)i, and to provide information that may motivate the audience to adopt the recommendation, as in (3)ii.[4] The plan parameter $?p2$ would be instantiated with the proposition describing the recommended action.[5] The *Motivate* plan constraint of the operator would instantiate the plan variable $?p1$ with the proposition expressed in (3)ii. In our current system, the search for a proposition satisfying a constraint such as the *Motivate* constraint in the example is performed by accessing a database created by a domain-specific data analysis component. For example, in our current application domain the data analysis component analyzes transportation schedules and records features that may be of interest to the user.

Propositions such as $?p2$ and $?p1$ are represented in a RQFOL (first-order logic with restricted quantification). RQFOL has been used for representing the meaning of natural language queries involving complex referring expressions [Webber1983, Woods1983]. In addition to providing a powerful, compositional representation scheme for the complex descriptions occurring in our domain,

[3] The tasks are called *logical tasks* since they do not presuppose a particular graphic design. All design decisions such as the type of graphic (e.g., a bar chart) or specific properties of a graphic (e.g., the choice of horizontal as opposed to vertical bars) are made by SAGE.

[4] We recognize that the intended effect of the plan may not be achieved. Dealing with plan failure is an area of future research.

[5] By convention, symbols beginning with $ are internal identifiers, and symbols beginning with ? are plan variables.

```
(define (action Recommend-act)
  :parameters (?p2)
  :primitive nil
  :effect
  ((accepted-recommendation $User ?p2))
```

```
(define (decomposition Recommend-act)
  :constraints
  ((Motivate ?p1 ?p2)
   (Proposition ?p1 ?main-pred1 ?refs1)
   (Proposition ?p2 ?main-pred2 ?refs2))
  :steps
  ((Recommend ?main-pred2 ?refs2)
   (Convey-relation Motivation ?p1 ?p2)
   (Assert ?main-pred1 ?refs1)))
```

Figure 2: Plan Operator for Discourse Strategy

RQFOL distinguishes information about discourse referents from the main predication of an expression. For example, the *Proposition* plan constraint of the operator in Figure 2, makes use of the RQFOL representation of $?p1$ to extract information with which to instantiate the plan variables *?main-pred1* and *?refs1* with the main predication of $?p1$ and a list describing the discourse entities [Webber1983] evoked or accessed by use of *?main-pred1*, respectively. (The significance to presentation generation of the distinction between the main predication and information about discourse referents is discussed in [Green et al.1998].)

The step of the operator shown in Figure 2 underlying (3)ii is an *Assert* action. In general, *Assert(?prop,?refs)* is defined as *the System asserts ?prop to the User*, where *?refs* is a list specifying all discourse entities evoked or accessed by use of *?prop*.[6] Discourse entities are specified in the list either by an internal identifier (an identifier referring to a database object) or by descriptions stated as RQFOL expressions.

For example, consider the *Assert* action underlying (3)ii, which can be represented as follows:[7]

```
Assert(serves($RI, d2),
   (($RI, ?d1),
    (d2, (for the c:city(c) has-population(c,d3)))
    (d3, (for the y:int(y) max(y,d4))),
    (d4, (for set z:int(z) has-population(d5, z))),
    (d5, (for all w:city(w) in(w,$SOMERSET)))
    ($SOMERSET, ?d6)))
```

The variable *?prop* has been instantiated with *serves($RI,d2)*, where *$RI* and *d2* are discourse entities; the variable *?refs* is instantiated with a list specifying six discourse entities:

- $\$RI$, a discourse entity (identified by the internal identifier of the database object which it is intended to represent) representing Realtors Inc.,[8]
- *d2*, described as the city with a population of *d3*,
- *d3*, described as the maximum value of *d4*,
- *d4*, described as a set of population values derived from *d5*,
- *d5*, described as all cities in $\$SOMERSET$, and
- *$SOMERSET*, the discourse entity/database object representing Somerset County.

[6]The order of list elements in *?refs* is used to represent quantifier scope.

[7]For readability, the example is presented in RQFOL-style syntax rather than the LISP-style syntax actually used in our implementation.

[8]The unbound variables *?d1* and *?d6* associated with $\$RI$ and $\$SOMERSET$ indicate that no descriptions have been associated with them by the planner.

```
(define (action Assert)
  :parameters (?prop ?refs)
  :primitive nil
  :effect
  ((bel $User ?prop ?refs)))
```

```
(define (decomposition Assert)
  :constraints
  ((get-id-list ?prop ?refs ?id-list)
   (get-desc-list ?prop ?refs ?desc-list))
  :steps
  ((Predicate ?prop)
   (forall ?id in ?id-list
        (Activate-ko ?id))
   (forall ?desc in ?desc-list
        (Activate-as ?desc ?refs))))
```

Figure 3: Plan Operator for Assert

Figure 3 shows the definition of an abstract *Assert* action and a simplified version of its decomposition. An *Assert* may be decomposed into three types of subactions. *Predicate* is used to describe an event independently of the things that play a role in that event.*Activate-ko* is a primitive action used to refer to an **object**, i.e., this corresponds to the referential use of a description. To achieve the effect of this action, the text and graphics generators are free to select any device that will enable the user to identify the object (subject to pragmatically appropriate identification constraints [Appelt and Kronfeld1987]). In other words, since the function of the description is purely referential, its content does not contribute directly to the presentation's goals and thus is not represented in the plan. *Activate-as* is used to refer to a discourse entity **as** the object fitting the description provided, i.e., this corresponds to the attributive use of a description. An *Activate-as* may itself be decomposed into these three types of subactions.

During hierachical planning, the constraints of the *Assert* decomposition operator (shown in Figure 3) are used to instantiate the plan variables *?id-list* and *?desc-list*. In the *forall* step of the operator, an *Activate-ko* and *Activate-as* action is created for each element of *?id-list* and *?desc-list*, respectively. E.g., for the *Assert* shown above representing (3)ii, the *?id-list* would contain the identifiers *$RI* and *$SOMERSET*, and *?desc-list* would include the descriptions of *d2* through *d5*. Then, the *Assert* shown above would be partly decomposed into attributive and referential communicative actions as follows: *$RI* is the object of an *Activate-ko* act, and *d2* is decomposed into an *Activate-as* act describing *d2*, which in turn is decomposed into an *Activate-as* describing *d3*, and so on, ending with an *Activate-ko* to enable the audience to identify *$SOMERSET*. In general, a complex attributive description may contain one or more *Activate-ko* acts. That is, our representation scheme supports the composition of descriptions for attributive use from subcomponents whose use may be attributive or referential. Thus, in this example, *$SOMERSET* could be described in a number of ways, e.g., *Somerset County* or *the county on the eastern side of Westmoreland County*.

To summarize the process of generating attributive descriptions in our approach, discourse strategies such as *Recommend-act* (shown in Figure 2) determine content selection as well as whether the selected information will be presented as part of the main predication or as part of an attributive description. The illocutionary act operators (e.g., *Assert*) and *Activate-as* operator further decompose any descriptions into *Activate-as* and *Activate-ko* acts. Thus, the system's intentions are represented in the presentation plan, enabling appropriate text and graphics to be generated. For example, because the information associated with *d2* (*the city with the largest population in Somerset county*) is part of the above plan, the graphic generator will attempt to produce a graphic such as (e) in Figure 1 that will enable the user to see that the agency serving the town with the largest population is Realtors Inc. Without such a specification in the plan, a graphic might be designed showing only that Realtors Inc. serves Berlin, or worse, that Realtors Inc. serves the city with the worst pollution in Somerset County. (For examples of how different communicative intentions can be distinguished in graphics see [Green et al.1998].)

6 Related Work

[Kronfeld1986, Kronfeld1990] distinguishes three independent aspects of the referential-attributive distinction, discusses the significance of the distinction for a computational model of reference, and describes how attributive descriptions may result in conversational implicatures [Grice1975]. The implications of the referential-attributive distinction for centering theory are discussed in [Grosz et al.1983]. [Appelt and Kronfeld1987] provides a formal theory that derives the effects of referring actions.

Previous integrated text and graphic generation systems, e.g., [Fasciano and Lapalme1996, Feiner and McKeown1991, Maybury1991, Wahlster et al.1993] have not attempted to perform task-based design of graphics as in our approach. Previous work on natural language reference in multimedia generation [Andre and Rist1994, McKeown et al.1992] has focused on coordination of pictorial and textual references to concrete objects and to actions to be performed on the objects, and on generating references to the presentation itself. Previous work on reference in sentence generation, e.g., [Appelt1985, Dale1992, Dale and Reiter1995, Heeman and Hirst1995, Horacek1997, Stone and Doran1997], has not addressed the referential-attributive distinction. [Elhadad1992] describes a representation scheme for specifying complex noun phrases, in which a set can be described either by its extension or intension. However, this distinction is independent of the referential-attributive distinction, since the same noun phrase can be used with either intention.

7 Conclusion

We have described a media-independent, compositional, plan-based approach to generating attributive descriptions for use in integrated text and graphics generation. An *attributive* description's main function is to convey information directly contributing to the communicative goals of a discourse. In our architecture, uses of referential and attributive descriptions are represented as two distinct types of communicative acts in a media-independent plan. It is particularly important to distinguish the two types of acts, since they have different consequences for dialogue followup behavior, text generation, and graphic design.

8 Acknowledgments

This project was supported by DARPA, contract DAA-1593K0005.

References

[Andre and Rist1994] E. Andre and T. Rist. 1994. Referring to world objects with text and pictures. In *Proceedings of COLING*, pages 530–34.

[Appelt and Kronfeld1987] D. Appelt and A. Kronfeld. 1987. A computational model of referring. In *Proceedings of The 10th IJCAI*, pages 640–647, Milan, Italy.

[Appelt1985] Douglas E. Appelt. 1985. Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33.

[Beshers and Feiner1993] C. Beshers and S. Feiner. 1993. AutoVisual: Rule-based design of interactive multivariate visualization. *IEEE Computer Graphics and Applications*, pages 41–49, July.

[Casner1991] S. M. Casner. 1991. A task-analytic approach to the automated design of information graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, April.

[Chuah et al.1995] Mei C. Chuah, Steven F. Roth, John Kolojejchick, Joe Mattis, and Octavio Juarez. 1995. SageBook: Searching data-graphics by content. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '95)*, pages 338–345. ACM/SIGCHI.

[Dale and Reiter1995] R. Dale and E. Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

[Dale1992] R. Dale. 1992. *Generating Referring Expressions: Building Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, Massachusetts.

[Donnellan1977] K. Donnellan. 1977. Reference and definite descriptions. In Stephen P. Schwartz, editor, *Naming, Necessity, and Natural Kinds*, pages 42–65. Cornell University Press, Ithaca.

[Elhadad and Robin1996] M. Elhadad and J. Robin. 1996. An overview of SURGE: A reusable comprehensive syntactic realization component. Technical Report Technical Report 96-03, Dept of Mathematics and Computer Science, Ben Gurion University, Beer Sheva, Israel.

[Elhadad1992] M. Elhadad. 1992. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. Ph.D. thesis, Columbia University.

[Fasciano and Lapalme1996] M. Fasciano and G. Lapalme. 1996. PostGraphe: a System for the Generation of Statistical Graphics and Text. In *Proceedings of the 8th International Natural Language Generation Workshop*, pages 51–60, Sussex, UK, June.

[Feiner and McKeown1991] S. Feiner and K. McKeown. 1991. Automating the generation of coordinated multimedia explanations. *IEEE Computer*, 24(10):33–40, October.

[Green et al.1998] Nancy Green, Giuseppe Carenini, Stephan Kerpedjiev, Steven Roth, and Johanna Moore. 1998. A media-independent content language for integrated text and graphics generation. In *Proceedings of the Workshop on Content Visualization and Intermedia Representations (CVIR'98) of the 17th International Conference on Computational Linguistics (COLING '98) and the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*. Coling-ACL'98 and ACL SIGMEDIA. To appear.

[Grice1975] H. Paul Grice. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics III: Speech Acts*, pages 41–58. Academic Press, New York.

[Grosz et al.1983] B. Grosz, A.K. Joshi, and S. Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting*, pages 44–50, Cambridge, Mass. Association for Computational Linguistics.

[Heeman and Hirst1995] P. Heeman and G. Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21(3):351–382.

[Horacek1997] H. Horacek. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 206–213, Madrid, Spain.

[Kerpedjiev et al.1997] Stephan Kerpedjiev, Giuseppe Carenini, Steven F. Roth, and Johanna D. Moore. 1997. Integrating planning and task-based design for multimedia presentation. In *International Conference on Intelligent User Interfaces (IUI '97)*, pages 145–152. Association for Computing Machinery.

[Kerpedjiev et al.1998] Stephan Kerpedjiev, Giuseppe Carenini, Nancy Green, Steven F. Roth, and Johanna D. Moore. 1998. Saying it in graphics: from intentions to visualizations. In *Proceedings of the Symposium on Information Visualization (InfoVis '98)*. IEEE Computer Society Technical Committee on Computer Graphics. To appear.

[Kronfeld1986] Amichai Kronfeld. 1986. Donnellan's distinction and a computational model of reference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 186–191.

[Kronfeld1990] Amichai Kronfeld. 1990. *Reference and computation: An essay in applied philosophy of language*. Cambridge University Press, Cambridge.

[Maybury1991] Mark T. Maybury. 1991. Planning multimedia explanations using communicative acts. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 61–66, July.

[McKeown et al.1992] K. McKeown, S. Feiner, J. Robin, D. Seligmann, and M. Tanenblatt. 1992. Generating cross-references for multimedia explanation. In *Proceedings of AAAI*, pages 9–16.

[Moore1995] J. D. Moore. 1995. *Participating in Explanatory Dialogues*. MIT Press.

[Roth and Mattis1990] S.F. Roth and J. Mattis. 1990. Data characterization for intelligent graphics presentation. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '90)*, pages 193–200.

[Roth et al.1994] Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '94)*, pages 112–117.

[Searle1970] John R. Searle. 1970. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, London, England.

[Stone and Doran1997] M. Stone and C. Doran. 1997. Sentence planning as description using tree adjoining grammar. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 198–205, Madrid, Spain.

[Wahlster et al.1993] W. Wahlster, E. Andre, W. Finkler, H.-J. Profitlich, and T. Rist. 1993. Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, 63:387–427.

[Webber1983] Bonnie L. Webber. 1983. So what can we talk about now? In B. Grosz, K. S. Jones, and B. L. Webber, editors, *Readings in Natural Language Processing*. Morgan Kaufmann, Los Altos, California.

[Woods1983] W. Woods. 1983. Semantics and quantification in natural language question answering. In B. Grosz, K. S. Jones, and B. L. Webber, editors, *Readings in Natural Language Processing*. Morgan Kaufmann, Los Altos, California.

[Young1994] Michael R. Young. 1994. A developer's guide to the Longbow discourse planning system. Technical Report ISP TR Number: 94-4, University of Pittsburgh, Intelligent Systems Program.