UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

## Title
Generating Temporal Expressions in Natural Language

Permalink
https://escholarship.org/uc/item/21996541

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 11(0)

Author
Forster, David R.

Publication Date
1989

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

# Generating Temporal Expressions in Natural Language

David R. Forster
Department of Computer and Information Science
University of Massachusetts

**Abstract**

We explore the problem of generating natural language temporal expressions and show that it is amenable to solution by the application of hierarchical constraint propagation. Constraints derived either directly or indirectly (via transformations) from client data are propagated over the hierarchical structure provided by syntactic templates and are required to be consistent at any given node. Multiple sources of constraints must be used to achieve lexical selection of a single item.

## INTRODUCTION

Consider an application program which schedules meetings on a calendar. Typically, such a program would need to produce sentences such as "*You will meet John Smith at 2:00 on Tuesday to discuss widgets.*" and "*You won't need to see Smith again today. You already saw him at ten [today].*" No generator, past or present, would be able to deal with all aspects of the generation of these sentences, because a complete model of tense and temporal adverbials is lacking. Linguistic models lack a connection to time units (e.g., minutes, hours, days) and common-sense knowledge in general, and also lack an acceptable mechanism for co-ordinating the processing of tense and temporal adverbials.

We will examine here a theory for the semantics of time units and temporal relations in natural language, and a companion method for processing this type of semantic information. The theory and method are being implemented as part of the semantic component for a text generation system (Forster, 1989). To reduce the work to a manageable level while retaining an interesting and useful body of data, we will concentrate on prepositional phrases serving as temporal adverbials (TPPs), in particular those using the prepositions *in*, *at*, and *on*.

Conventionally, *at* is used to refer to points in time or to intervals which may be treated as points, *on* is used to refer to days or (together with a specific day) to major parts of days, and *in* is used otherwise (Quirk et al., 1985:526-555). Most models for the semantics of these prepositional

This work was supported in part by the Air Force Systems Command, Rome Air Development Center, Griffiss AFB, New York, 13441 under contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC).

phrases depend on precedence or on containment relations, with no further detail attempted.[1]

There are several problems with the conventional analysis of TPPs. First, it is too vague; for example, no rules are given governing which intervals may be treated as points and which may not. Second, it is both too powerful and too weak — witness examples (1–5) below. Third, it doesn't even touch on the interactions of TPPs with other mechanisms for expressing temporal relations, like those exhibited by examples (6–8) below.[2] Fourth, it does not link the objects being described with the descriptions in any useful way.[3]

1. *?at coffee break*
2. **at Canada Day/Independence Day*
3. **at summer*
4. *on the following evening/morning*
5. *#at 9 in January*
6. *#I saw him tomorrow.*
7. *#At ten I had seen him at ten.*
8. *?It will last [forever] at ten.*

We must draw on our common-sense knowledge of how these events differ to explain these observations. By examining differences between events which figure in examples of acceptable and unacceptable text, we should be able to determine the common-sense knowledge required to correctly generate temporal expressions.

## A COHERENT TREATMENT OF TIME

The problems with generating TPPs may be summarised as follows:

1. Certain temporal units require the use of specific prepositions. Where variations on these units are allowed (e.g. *Christmas* vs. *Christmas Day*), the prepositions allowed for the variant usually differ from those allowed for the "root" form.
2. The agreement of tense and temporal adverbials must be ensured in some way.
3. Time intervals can sometimes be treated as points in time.
4. Prepositional phrases detailing different levels of time units describing the same event may be combined, but levels may not be skipped, unless they are clear from context (cf. example (5) above).

The solution I propose has four parts:

[1] **Precedence** simply means that one time is before another, i.e. one precedes another. **Containment** simply refers to the time of a particular event being in the set of times contained by another time or event.

[2] In (6), the tense indicates that $T_s > T_e$, whereas the adverbial indicates that $T_s < T_e$. In (7), the aspect indicates $T_e < T_r$, whereas the adverbial indicates that $T_e = T_r$. In (8), the verb "*last*" asserts that some state holds for a period of time, whereas the adverbial indicates that the state holds only for an instant. Several other forms of interaction are also known.

[3] By this I mean that nothing is specified about the mapping from application program data to linguistic data or about acquiring the data from the application program or even about what sort of data is required. Insofar as the conventional analysis is purely linguistic, performing this mapping may be judged to be beyond the scope of the analysis. For our purposes, however, this link must also be established.

1. Associate constraints with each word and phrase describing how the word or phrase can be used and what it refers to (e.g., "*at*" used to pick out a point in time, "*Tuesday*" referring to a one day long interval).
2. Base the constraints on common-sense knowledge.
3. Allow the constraints to propagate in the syntax tree for the sentence.
4. Require constraints that meet at a node to be consistent.

In addition to this, the client (the program requesting the generation of text) must be required to make some of the decisions which can be expressed in non-linguistic terms, and to bundle some of its information (e.g., the level of detail it requires for a description). If insufficient information is available in the initial generation request made by the client, then it should be possible to query the client about its needs or preferences.[4]

This solution remedies the problems with generation by

1. forcing the use of the correct prepositions and the agreement of tense and temporal adverbials by propagating constraints and enforcing constraint satisfaction
2. enabling the use of variations in phrasing, and enabling the treatment of intervals as points, by basing the constraints on the additional common-sense attributes of described objects.

The solution remedies weaknesses of the current theoretical model by

1. modelling interactions explicitly
2. defining links between client objects (representing real-world objects) and linguistic objects
3. attending to detail to a degree not practicable for a theoretical system, but required for the implementation of a processing model

#### Time and Common-Sense Knowledge

In light of the data, it is obvious that the choice of a preposition must be in accord with the nature of the temporal datum being described, and equally obvious that more than just the 'size' of that datum, or precedence and containment relations with other data are important. At least the following attributes appear necessary:

**Regularity** with which an event occurs: breakfast is such a regular event it may be considered to be a useful reference point, thus allowing its treatment as a point in "*at breakfast.*"

[4]It could be argued that the client should include all the information it wants expressed in the initial generation request, since queries made later merely represent a delay in collecting needed information. This argument ignores the fact that the mechanism for bundling the information may well assume things about either the surface structure or the deep structure of sentences, and therefore force the client to have knowledge of language.

**Significance** of the event: Christmas is such an important event in both the church and secular calendars that we treat it as a point, allowing "*at Christmas*."

**Duration** of an interval: the duration of a weekend has the same order of magnitude as a day, thus possibly allowing its treatment as a day in *on weekends*.

**Perception** of time units: days have clearly defined boundaries, determined by a natural external source (the rising and setting of the sun), thus allowing us to conceive of them as somehow fixed or solid, and by extension, as things "on" which events can in some sense be "put," in analogy to spatial perception.

**Granularity** of time units:[5] the use of adverbs, such as *just*, is clearly a sign that there is an interval within which events are considered to be recent. Granularity is the size of this interval. Further research may show that a more complex notion, allowing different classes of recency, is required.[6]

### Hierarchical Constraint Propagation

Stefik (1981) outlines a method for planning based on constraint satisfaction in a hierarchy, where a rough hierarchical plan is refined by using constraints to determine valid variable values in plan steps, and to reason out further constraints, which are propagated throughout the hierarchy of the plan. Stefik notes that constraint propagation is useful only when the problem involves loosely coupled subsystems. Natural language generation is an ideal candidate for a solution of this kind. The selection process for one word or phrase is loosely coupled with the selection of others. Realisations for words and phrases are selected on the basis of satisfied constraints. The constraints are provided by word meanings or transformations on inherited constraints. The hierarchy is provided by the syntactic structure of the sentence.

Figure 1 shows an example of hierarchical constraint propagation, concentrating on temporal expressions. Let us examine in turn the generation of each temporal expression. "*At*" produces the constraint (time-nature :point), which is propagated to the containing PP at (C) in the figure. "*Nine*" provides the information ((time-nature :point) (time #<TIME see>)), which is propagated to the same PP at (D). The generation of "*at nine*" proceeds by satisfying the point requirement of *at* with the point nature of the hour *nine*. "*On the seventeenth*" is generated similarly (F&G),

[5]This is distinct from Hobbs' (1985) use of the term granularity.

[6]Three such classes which suggest themselves are: remote, recent, and immediate, corresponding to sentences like:

1. "*Bill left a long time ago.*"
2. "*John left today.*"
3. "*Tom just left.*"

It is not clear whether the size of objects in one class is a function of the size of an object in another class, or whether the sizes are unrelated. If these sentences were spoken in this order in close succession, we could imagine a situation in which the departure in (1) was a few hours prior to the time of speech, and the departure in (3) was just seconds prior. Thus, we could have the following correspondences: immediate: $\Delta t \leq minutes$, recent: $\Delta t \leq hours$, and remote: $\Delta t > hours$. By the same token, we could imagine a situation in which the departure in (1) was many years prior to the time of speech, while at the same time maintaining that in (3) was just seconds prior. The given correspondences would remain the same, except for remote being $\Delta t > years$.

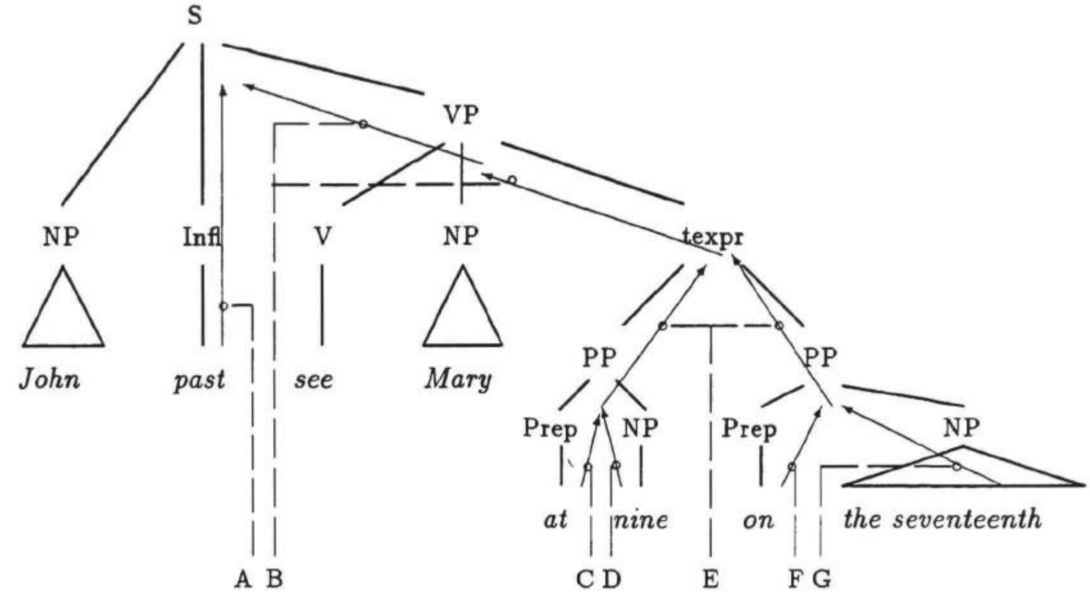


Figure 1: Hierarchical constraint propagation and temporal expressions

with the constraint from *on* being `(time-nature :day-interval)` and the information from *the seventeenth* being `((time-nature :day-interval) (time #<TIME see>))`.[7] Of the information collected at each of these two nodes, only `(time #<TIME see>)` is propagated up to 'texpr' (E). These constraints are combined and propagated ultimately to the S-node (B), to be combined with the constraint derived from the tense to be used (A).

This technique can be extended to the curious phrases examined in section 1. *At* could be modified to allow a holiday season (as opposed to a 'climatic' season, such as summer) as well as a point in time. If the client identifies the time it has provided as "*Christmas*", and also declares it to be a holiday season, then the phrase "*at Christmas*" may be produced, as required by the observations.

## IMPLEMENTATION

The overall system includes a client program (currently a calendar scheduling program), a semantic component (GNOMON), and a syntactic component (MUMBLE (McDonald, 1983)). GNOMON uses hierarchical constraint propagation to choose applicable realisations and lexical items based on constraints derived from the realisations themselves and from context. The hierarchy mirrors the syntax tree, with sets of possible realisations at each node. A possible realisation provides information about the syntactic form to be used to realise a particular phrase, the constraints on its use, and the effect the form will have on context. The syntactic form is represented (recursively) by a form consisting of directives for the syntactic component and zero or more uninstantiated forms.

[7]The time referenced here is some client structure representing the time of the 'seeing' event, i.e. nine o'clock on the seventeenth. We refer to it here as TIME see to avoid confusion.

Four operations on the hierarchy and its members are defined: An uninstantiated form in the syntactic form of a possible realisation may be **expanded** by finding possible realisations for it. Constraints may be **propagated** to other parts of the realisation tree, possibly after the application of some reasoning. A set of possible realisations may be **narrowed** by the application of constraints, possibly resulting in the instantiation of one of them. Finally, a set of possible realisations may be **filtered**, drawing on non-linguistic criteria (e.g., brevity) for the selection.[8] GNOMON begins by selecting a top-level underspecified form for realisation and creating a queue containing it alone. GNOMON then processes the entries in the queue, passing on the final syntactic form to the syntactic component and recording the effects on global context when the queue is empty. The exact action to be taken when processing a queue entry depends on advice from a set of higher level constraints. These constraints are intended to reflect conventions in speech which lie outside of semantics and syntax, such as Gricean maxims. Though the actual constraints to be used are in a state of flux, we can assume for the time being that they include:

1. uninstantiated forms are instantiated whenever only one realization is known for it
2. constraints are propagated whenever they are introduced at a node[9]
3. possible realizations found to be inconsistent with already established constraints are automatically filtered out
4. possible realizations are always filtered to prefer anaphoric references, if present
5. the leftmost unexpanded node is selected for expansion.

Processing ends when no forms needing expansion remain. The form is then given to the syntactic component for processing, after which the effects on global context are recorded.

## RELATED WORK

Matthiessen (1984) is an excellent survey of work on the use of tense to express the temporal relations underlying an utterance. Pustejovsky (1987) presents a novel theory of aspect which allows the internal event structure of a verb to be modelled. Many linguists (Prior, 1967; Reichenbach, 1947; Montague, 1974; Bennett & Partee, 1978; Dowty, 1979; Johnson, 1981) have suggested models for tense. Not all of them considered the interaction of their model with temporal adverbials. All gloss over some details of the representation. One of the more complete works is that of Dowty (1979), which presents a framework and definitions for a substantial fragment of English, and which discusses temporal adverbials and interactions with tense in some detail.

Hierarchical constraint propagation is related to functional unification grammar (Kay, 1984; Shieber, 1986) but uses constraints instead of features, and consistency checks instead of

[8] Note that filtering is not intended as a stopgap for short-comings of the grammar or the constraint propagation mechanism, but rather as a model of extra-linguistic influences on lexical choice. For example, it may be invoked when the speaker is pressed for time.

[9] More correctly, constraints are propagated whenever they are found to be common to all possible realizations attached to a node.

unification, resulting in general and powerful framework. GNOMON should attain significantly improved computational complexity through judicious use of its filtering capabilities.

## CONCLUSIONS

We have explored the problem of generating temporal expressions and shown that it is amenable to solution by the application of hierarchical constraint propagation. Constraints derived either directly or indirectly (via transformations) from client data are propagated over the hierarchical structure provided by syntactic templates and are required to be consistent at every node. Multiple sources of constraints must be used to achieve lexical selection of a single item. Operations on the structure are controlled by higher-order constraints.

This approach allows the proper handling of the temporal prepositional phrases by the application of common-sense knowledge and in particular the simultaneous specification of tense and multiple temporal adverbials. Furthermore, it allows the modelling of non-linguistic interactions within the grammar, a closer integration of common-sense data with 'meaning,' and a better understanding of the interface and underlying knowledge needed by client programs to take advantage of natural language generation programs.

Further work will involve extending the grammar to deal with a wider range of temporal expressions and the application of common-sense knowledge to other types of expressions.

## ACKNOWLEDGEMENTS

Thanks are due to James Pustejovsky, Emmon Bach, Victor Lesser, Beverly Woolf, Robin Popplestone, Penni Sibun, and Scott Anderson and the NLG discussion group at UMass who have offered many useful comments on the work.

## REFERENCES

Bennett, M., & Partee, B. H. (1978) *Toward the Logic of Tense and Aspect in English,* Indiana University Linguistics Club, Bloomington, Indiana.

Dowty, D. (1979) *Word Meaning and Montague Grammar*, Reidel.

Forster, D. R. (1989) *Time and Natural Language Generation,* Technical Report 89-01, Department of Computer and Information Sciences, University of Massachusetts, Amherst.

Grosz, B., Sparck Jones, K. & Webber, B. L. (eds.) (1986) *Readings in Natural Language Processing,* Kaufmann.

Hobbs, J. R. (1985) *Granularity,* IJCAI-85, 1985, pp. 432-435.

Johnson, M. R. (1981) *A Unified Theory of Tense and Aspect*, in Tedeschi, P., & Zaenen A. (eds.) Tense and Aspect, Syntax and Semantics, v. 14, Academic Press, pp. 145-176.

Kay, M. (1984) *Functional Unification Grammar: A Formalism for Machine Translation*, COLING 84. pp. 75-78.

Matthiessen, C. (1984) *Choosing Tense in English*, ISI/RR-84-143.

McDonald, D. D. (1983) *Description Directed Control*, Computers and Mathematics 9(1), 1983, pp. 111-130, also in Grosz, Sparck Jones & Webber (eds.), 1986, pp. 519-537.

Montague, R. (1974) *The Proper Treatment of Quantification in Ordinary English*, in Thomason (ed.), Formal Philosophy, Yale, pp. 247-270.

Prior, A. N. (1967) *Past, Present, and Future*, Oxford University Press.

Pustejovsky, J. (1987) *An Event Structure for Lexical Semantics*, TR Brandeis University Computer Science Department.

Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985) *A Comprehensive Grammar of the English Language*, Longman.

Reichenbach, H. (1947) *Elements of Symbolic Logic*, Macmillan, republished by Dover, 1980.

Shieber, S. M. (1986) *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Notes Number 4.

Stefik, M. (1981) *Planning with Constraints (MOLGEN: Part 1)*, AI 16(2), pp. 111-140.