1 Dialogue and Compound Contributions

Matthew Purver[†], Julian Hough[†] and Eleni Gregoromichelaki[*]
[†]Queen Mary University of London and [*]King's College London

1.1 Introduction

In this chapter, we examine the phenomenon of *compound contributions* (CCs) and the implications for NLG in interactive systems. Compound contributions are contributions in dialogue which continue or complete an earlier contribution, thus resulting in a single syntactic or semantic (propositional) unit built across multiple contributions, provided by one speaker or more than one (1.1). The term as used here therefore includes more specific cases that have been referred to as *expansions*, *(collaborative) completions*, and *split* or *shared utterances*.

(1.1) (Friends of the Earth club meeting:)
A: *So what is that? Is that er . . . booklet or something?*
B: *It's a* [[*book*]]
C: [[*Book*]]
B: *Just . . .* [[*talking about al– you know alternative*]]
D: [[*On erm . . . renewable yeah*]]
B: *energy really I think*
A: *Yeah* [*BNC D97 2038-2044*][1]

As we discuss in Section 1.2, a dialogue agent that processes or takes part in CCs must change between parsing (NLU) and generation (NLG), possibly many times, while keeping track of the representation being constructed. This imposes some strong requirements on the nature of NLG in interactive systems, in terms both of NLG incrementality and of NLG-NLU interdependency. As we explain in Section 1.3, current approaches to NLG, while exhibiting incrementality to substantial degrees for various reasons, do not yet entirely satisfy these requirements. In Sections 1.4 and 1.5 we outline one possible approach to NLG that is compatible with CCs; it uses the *Dynamic Syntax* (DS) grammatical framework with *Type Theory with Records* (TTR). We explain how DS-TTR might be incorporated into an interactive system. In Section 1.6 we outline how this approach can be used in principle without recourse to recognising or modelling interlocutors' intentions, and how it is compatible with emerging empirical evidence about alignment between dialogue participants.

[1] Examples labelled *BNC* are taken from the British National Corpus (Burnard, 2000).

1.2 Compound Contributions

1.2.1 Introduction

Interlocutors very straightforwardly shift between the roles of listener and speaker, without necessarily waiting for sentences to end. This results in the phenomenon we describe here as *compound contributions* (CCs): syntactic or semantic units made up of multiple dialogue contributions, possibly provided by multiple speakers. In (1.1) above, B begins a sentence; C interrupts (or takes advantage of a pause) to clarify a constituent, but B then continues to extend his or her initial contribution seamlessly. Again, D interrupts to offer a correction; B can presumably process this, and continues with a final segment – which may be completing either B's original or D's corrected version of the contribution. We can see C and D's contributions, as well as B's continuations and completions, as examples of the same general CC phenomenon: in each case, the new contribution is continuing (optionally including editing or repair) an antecedent that may well be incomplete, and may well have been produced by another speaker.

CCs can take many forms. Conversation Analysis research has paid attention to some of these, in particular noting the distinction between expansions (contributions that add additional material to an already complete antecedent (1.2)) and completions (contributions that complete an incomplete antecedent (1.3)). A range of characteristic structural patterns for these phenomena, and the corresponding speaker transition points, have been observed: expansions often involve optional adjuncts such as sentence relatives (1.2); and completions often involve patterns such as IF-THEN (1.3) or occur opportunistically after pauses (1.4) (see (Ono and Thompson, 1993, Lerner, 1991, 1996, 2004, Rühlemann and McCarthy, 2007), among others).

(1.2) A: *profit for the group is a hundred and ninety thousand pounds.*
B: *Which is superb.* [*BNC FUK 2460-2461*]

(1.3) A: *Before that then if they were ill*
B: *They get nothing.* [*BNC H5H 110-111*]

(1.4) A: *Well I do know last week thet=uh Al was certainly very <pause 0.5>*
B: *pissed off* [*Lerner (1996, p. 260)*]

Clearly, examples such as these impose interesting requirements for NLG. Agent B must generate a contribution that takes into account the possibly incomplete contribution from agent A, both in syntactic terms (continuing in a grammatical fashion) and in semantic terms (continuing in a coherent and/or plausible way). And corpus studies (Szczepek, 2000, Skuplik, 1999, Purver et al., 2009) suggest that these are not isolated examples: CCs are common in both task-oriented and general open-domain dialogue, with around 3% of contributions in dialogue continuing some other speaker's material (Howes et al., 2011).

1.2.2 Data

The regular patterns of (1.2–1.4) already show that agents can continue or extend utterances across speaker and/or turn boundaries; but these patterns are by no means the only possibilities. In this section, we review some other possible CC forms, using data from (Purver et al., 2009, Gregoromichelaki et al., 2011), and note the rather strict requirements they impose.

1.2.2.1 Incrementality

In dialogue, participants ground each other's contributions (Allen et al., 2001) through backchannels like *yeah, mhm*, etc. This is very often done at incremental points within a sentence: the initial listener shifts briefly to become the speaker and produce a grounding utterance (with the initial speaker briefly becoming a listener to notice and process it), and roles then revert to the original:

(1.5) A: *Push it, when you want it, just push it* [pause] *up there*
B: *Yeah.*
A: *so it comes out.* [*BNC KSR 30-32*]

(1.6) A: *So if you start at the centre* [pause] *and draw a line and mark off seventy two degrees,*
B: *Mm.*
A: *and then mark off another seventy two degrees and another seventy two degrees and another seventy two degrees and join the ends,*
B: *Yeah.*
A: *you'll end up with a regular pentagon.* [*BNC KND 160-164*]

We see these as examples of CCs in that the overall sentential content is spread across multiple speaker turns (although here, all by the same speaker). NLG processes must therefore be interruptible at the speaker transition points, and able to resume later.

In addition, the speaker must also be able to process and understand the grounding contribution – at least, to decide whether it gives positive or negative feedback. In (1.5 –1.6) above, one might argue that this requires little in the way of understanding; however, the grounding contribution may provide or require extra information which must be processed in the context of the partial contribution produced so far:

(1.7) A: *And er they X-rayed me, and took a urine sample, took a blood sample. Er, the doctor*
B: *Chorlton?*
A: *Chorlton, mhm, he examined me, erm, he, he said now they were on about a slide* [unclear] *on my heart.* [*BNC KPY 1005-1008*]

(1.8) (Friends of the Earth club meeting (repeat of (1.1) above):)
A: *So what is that? Is that er . . . booklet or something?*
B: *It's a* [[*book*]]

C: [[*Book*]]
B: *Just . . .* [[*talking about al– you know alternative*]]
D: [[*On erm . . . renewable yeah*]]
B: *energy really I think*
A: *Yeah* [*BNC D97 2038-2044*]

Contributions by B in (1.7) and by C and D in (1.8) clarify, repair or extend the partial utterances, with the clarification apparently becoming absorbed into the final, collaboratively derived, content. Processing these contributions must require not only suspending the initial speaker's NLG process, but providing the partial representations it provides to his or her NLU processes, allowing for understanding and evaluation of the requested confirmation or correction. As NLG then continues, it must do so from the newly clarified or corrected representation, including content from all contributions so far.

Note that (1.7) shows that the speaker transition point can come even in the middle of an emergent clause; and the transitions around D's contribution in (1.8) occur within a noun phrase (between adjective and noun). Indeed, transitions within any kind of syntactic constituent seem to be possible, with little or no constraint on the possible position (Howes et al., 2011, Gregoromichelaki et al., 2011), suggesting that incremental processing must be just that – operating on a strictly word-by-word basis:

(1.9) A: [. . .] *whereas qualitative is* [pause] *you know what the actual variations*
B: *entails*
A: *entails. you know what the actual quality of the variations are.* [*BNC G4V 114-117*]

(1.10) A: *We need to put your name down. Even if that wasn't a*
B: *A proper* [[*conversation*]]
A: [[*a grunt*]]. [*BNC KDF 25-27*]

(1.11) A: *All the machinery was*
B: [[*All steam.*]][2]
A: [[*operated*]] *by steam* [*BNC H5G 177-179*]

(1.12) A: *I've got a scribble behind it, oh annual report I'd get that from.*
B: *Right.*
A: *And the total number of* [[*sixth form students in a division.*]]
B: [[*Sixth form students in a division.*]] *Right.* [*BNC H5D 123-127*]

1.2.2.2 Syntactic Dependencies

However, despite this apparent flexibility in transition point, syntactic dependencies seem to be preserved across the speaker transition:

[2] The [[brackets indicate overlapping speech between two subsequent utterances- i.e. here A's "operated" overlaps B's "All steam".

(1.13) (With smoke coming from the kitchen:)
A: *I'm afraid I burnt the kitchen ceiling*
B: *But have you*
A: *burned myself? Fortunately not.*

(1.14) A: *Do you know whether every waitress handed in*
B: *her tax forms?*
A: *or even any payslips?*

The negative polarity item *any* in A's final contribution (1.14) is licensed by the context set up in the initial partial antecedent; similarly the scope of A's quantifier *every* must include B's anaphoric *her*. In (1.13), A's reflexive pronoun depends on B's initial subject. The grammar, then, seems to be crucially involved in licensing CCs. However, this grammar cannot be one which licenses strings: the complete sentence gained by joining together B and A's CC in (1.13), *have you burned myself*, is not a grammatical string. Rather, semantic and contextual representations must be involved with the syntactic characterisations utilised to underpin the co-construction but not inducing a separate representation level.

1.2.2.3 Semantics and Intentionality

In many CC examples, the respondent appears to have guessed what they think was intended by the original speaker. These have been called *collaborative completions* (Poesio and Rieser, 2010):

(1.15) (Conversation from A and B, to C:)
A: *We're going to ...*
B: *Bristol, where Jo lives.*

(1.16) A: *Are you left or*
B: *Right-handed.*

Such examples must require that semantic representations are being created incrementally, making the partial meaning available at speaker transition in order to allow the continuing agent to make a correct guess. The process of inferring original intentions may be based on an agent's understanding of the extra-linguistic context or domain (see (Poesio and Rieser, 2010) and discussion below). But this is not the only possibility: as (1.17–1.18) show, such completions by no means need to be what the original speaker actually had in mind:

(1.17) Morse: *in any case the question was*
Suspect: *a VERY good question inspector* [*Morse, BBC radio 7*]

(1.18) Daughter: *Oh here dad, a good way to get those corners out*
Dad: *is to stick yer finger inside.*
Daughter: *well, that's one way.* [*from (Lerner, 1991)*]

In fact, such continuations can be completely the opposite of what the original speaker might have intended, as in what we will call *hostile continuations* or

devious suggestions which are nevertheless collaboratively constructed from a grammatical point of view:

(1.19) (A and B arguing:)
A: *In fact what this shows is*
B: *that you are an idiot*

(1.20) (A mother, B son:)
A: *This afternoon first you'll do your homework, then wash the dishes and then*
B: *you'll give me £10?*

Note, though, that even such examples show that syntactic matching is preserved, and suggest the availability of semantic representations in order to produce a continuation that is coherent (even if not calculated on the basis of attributed intentions).

1.2.3 Incremental Interpretation vs. Incremental Representation

It is clear, then, that a linguistic system which is to account for the data provided by CCs must be incremental in some sense: at apparently any point in a sentence, partial representations must be provided from the comprehension (NLU) to the production (NLG) facility, and vice versa. These processes must therefore be producing suitable representations *incrementally*; they must also be able to exchange them, requiring the quality of *reversibility*, in that representations available in interpretation should be available for generation too (see (Neumann, 1998), and below).

A pertinent question, then, is to what degree incrementality is required, and at which levels. In terms of interpretation, Milward (1991) points out the difference between a linguistic system's capacity for *strong incremental interpretation* and its ability to access and produce *incremental representations*. Strong incremental interpretation is defined as a system's ability to extract the maximal amount of information possible from an unfinished utterance as it is being produced, particularly the semantic dependencies of the informational content (*e.g.* a representation such as $\lambda x.like'(john', x)$ should be available after parsing *John likes*). Incremental representation, on the other hand, is defined as a representation being available for each substring of an utterance, but not necessarily including the dependencies between these substrings (*e.g.* a representation such as $john'$ being available after consuming *John* and then $john' * \lambda y.\lambda x.like'(y, x)$ being available after consuming *likes* as the following word).

Systems may exhibit only one of these different types of incrementality. This is perhaps most clear for the case of a system producing incremental representations but not yielding strict incremental interpretation – that is to say, a system which incrementally produces representations $\lambda y.\lambda x.like'(y, x)$ and $john'$, but does not carry out functional application to give the maximal possible semantic

information $\lambda x.like'(john', x)$. But the converse is also possible: another system might make the maximal interpretation for a partial utterance available incrementally, but if this is built up by adding to a semantic representation without maintaining lexical information – for example by the incremental updating of Discourse Representation Structures (DRS; for details see (Kamp and Reyle, 1993)) – it may not be possible to determine which word or sequence of words was responsible for which part of the semantic representation, and therefore the procedural or construction elements of the context may be irretrievable.

The evidence reviewed above, however, suggests that a successful model of CCs would need to incorporate both strong incremental interpretation *and* incremental representation, for each word uttered sequentially in a dialogue. Representations for substrings and their contributions are required for clarification and confirmation behaviour (1.7–1.8); and partial sentential meanings including semantic dependencies must be available for coherent, helpful (or otherwise) continuations to be suggested (1.17–1.20).

1.2.4 CCs and Intentions

However, incremental comprehension cannot be based primarily on guessing speaker intentions or recognising known discourse plans: for instance, it is not clear that in (1.17–1.20) the addressee has to have guessed the original speaker's (propositional) intention/plan before offering a continuation[3]. Moreover, speaker plans need not necessarily be fully formed before production: the assumption of fully-formed propositional intentions guiding production will predict that all the cases above where the continuation is not as expected, as in (1.17–1.20), would have to involve some kind of revision or backtracking on the part of the original speaker. But this is not a necessary assumption: as long as the speaker is licensed to operate with partial structures, he or she can start an utterance without a fully formed intention/plan as to how it will develop (as the psycholinguistic models in any case suggest) relying on feedback from the hearer to shape the utterance (Goodwin, 1979). The importance of feedback in co-constructing meaning in communication has been already documented at the propositional level (the level of speech acts) within Conversational Analysis (CA) (see *e.g.* (Schegloff, 2007)). However, it seems here that the same processes can operate sub-propositionally, but only relative to grammar models that allow the incremental, sub-sentential integration of cross-speaker productions.

[3] These are cases not addressed by DeVault et al. (2009), who otherwise offer a method for getting full interpretation as early as possible. Lascarides and Asher (2009), Asher and Lascarides (2008) also define a model of dialogue that partly sidesteps many of the issues raised in intention recognition. But, in adopting the essentially suprasentential remit of Segmented Discourse Representation Theory (SDRT), their model does not address the step-by-step incrementality required to model split-utterance phenomena.

1.2.5 CCs and Coordination

Importantly, phenomena such as (1.1–1.20) are not dysfunctional uses of language, unsuccessful acts of communication, performance issues involving repair, or deviant uses. If one were to set them aside as such, one would be left without an account of how people manage to understand what each other have said in these cases. In fact, it is now well documented that such "miscommunication" not only provides vital insights as to how language and communication operate (Schegloff, 1979), but also facilitates dialogue coordination: as Healey (2008) shows, the local processes involved in the detection and resolution of misalignments during interaction lead to significantly positive effects on measures of successful interactional outcomes (see also (Brennan and Schober, 2001)); and, as Saxton (1997) shows, in addition, such mechanisms, in the form of negative evidence and embedded repairs, crucially mediate language acquisition (see also (Goodwin, 1981, pp. 170–171)). Therefore, miscommunication and the specialised repair procedures made available by the structured linguistic and interactional resources available to interlocutors are the sole means that can guarantee intersubjectivity and coordination.

1.2.6 Implications for NLG

To summarize, the data presented above show that CCs impose some strong requirements on NLG (and indeed on NLU):

- Full word-by-word *incrementality*: NLG and NLU processes must both be able to begin and end at any syntactic point in a sentence (including within syntactic or semantic constituents).
- Strong *incremental interpretation*: an agent must be able to produce and access meaning representations for partial sentences on a word-by-word basis, to be able to determine a coherent, plausible or collaborative continuation.
- *Incremental representation*: an agent must be able to access the lexical, syntactic and semantic information contributed by the constituent parts processed so far, to process or account for clarifications and confirmations.
- *Incremental context*: agents must incrementally add to and read from context on a word-by-word basis, to account for cross-speaker anaphora and ellipsis and for changing references to participants.
- *Reversibility*, or perhaps better *interchangeability*: the partial representations (meaning and form) built by NLU at the point of speaker transition must be suitable for use by NLG as a starting point, and vice versa, preserving syntactic and semantic constraints across the boundary.
- *Extensibility*: the representations of meaning and form must be extendable, to allow the incorporation of extensions (adjuncts, clarifications etc.) even to complete antecedents.

As we shall see in the next section, previous and current work on incremental NLG has produced models that address many of these requirements, but not all; in subsequent sections, we then outline a possible approach that does.

1.3 Previous Work

In this section, we review existing research in incremental production and CCs, both from a psycholinguistic and from a computational perspective.

1.3.1 Psycholinguistic Research

The incrementality of on-line linguistic processing is now uncontroversial. Standard psycholinguistic models assume that language comprehension operates incrementally, with partial interpretations being built more or less on a word-by-word basis (see *e.g.* (Sturt and Crocker, 1996)). Language production has also been argued to be incremental (Kempen and Hoenkamp, 1987, Levelt, 1989, Ferreira, 1996), with evidence also coming from self-repairs and various types of speech errors (Levelt, 1983, van Wijk and Kempen, 1987).

Guhe (2007) further argues for the incremental conceptualisation of observed events in a visual scene. He uses this domain to propose a model of the incremental generation of preverbal messages, which in turn guides down-stream semantic and syntactic formulation. In the interleaving of planning, conceptual structuring of the message, syntactic structure construction and articulation, incremental models assume that information is processed as it becomes available, operating on minimal amounts of characterstic input to each phase of generation, reflecting the introspective observation that the end of a sentence is not planned when one starts to utter its beginning (Guhe et al., 2000). The evidence from CCs described above supports these processing claims, along with providing additional evidence for the ease of switching roles between incremental parsing and incremental generation during dialogue.

1.3.2 Incrementality in NLG

Early work on incremental NLG was motivated not only by the emerging psychological evidence, but also by attempts to improve user experience in natural language interfaces: systems that did not need to compile complete sentence plans before beginning surface realization could allow decreased response time to user utterances. Levelt (1989)'s concepts of the *conceptualisation* and *formulation* stages of language production lead to a more concrete computational distinction between the *tactical* and *strategic* stages of generation (Thompson, 1977), with the incremental passing of units between these becoming important. Parallel and distributed processing across modules stands in contrast to traditional

pipelined approaches to NLG (see *e.g.* (Reiter and Dale, 2000)), a shortcoming of generation architectures outlined perspicuously by De Smedt et al. (1996).

With formalisms such as Functional Unification Grammar (FUG; (Kay, 1985)) and Tree Adjoining Grammar (TAG; (Joshi, 1985)), researchers began to address incrementality explicitly. Kempen and Hoenkamp (1987) made the first notable attempt to implement an incremental generator, introducing their Incremental Procedural Grammar (IPG) model. Schematically, IPG was driven by parallel processes whereby a team of syntactic modules worked together on small parts of a sentence under construction, with the sole communication channel as a stack object (with different constituents loaded onto it), rather than the modules being controlled by a central constructing agent. This approach was consistent with emerging psycholinguistic theories that tree formation was simultaneously conceptually and lexically guided, and that production did not take place in a serial manner; it was capable of generating elliptical answers to questions and also some basic self-repairs.

De Smedt (1990) took incrementality a stage further, showing how developing the syntactic component of the formulation phase in detail could support cognitive claims, shedding light on lexical selection and memory limitations (De Smedt, 1991). De Smedt's Incremental Parallel Formulator (IPF) contained a further functional decomposition between grammatical and phonological encoding, meaning that syntactic processes determining surface form elements like word order and inflection could begin before the entire input for a sentence had been received.

Early incremental systems allowed input to be underspecified in the strategic component of the generator before the tactical component began realizing an utterance, paving the way for shorter response times in dialogue systems but without implementational evidence of such capability. It is worth noting the analogous situation in psycholinguistics: models including the functional decomposition of production stages, as described above, were influential in the autonomous processing camp of psycholinguistics; however, they did not extend to explaining the role of incremental linguistic processing in interaction.

1.3.3 Interleaving Parsing and Generation

In moving towards the requirements of an interactive system capable of dealing with CCs, notable work on interleaving generation with parsing in an incremental fashion came from (Neumann, 1994, Neumann and van Noord, 1994, Neumann, 1998), who showed how the two processes could be connected using a reversible grammar. The psychological motivation came mainly from Levelt (1989)'s concept of a feedback loop to parsing during generation for self-monitoring. The representations used by the parser and generator were explicitly reversible, based around *items*, pairs of logical forms (LFs – in this case, HPSG-like attribute-value matrices) and the corresponding strings.

Processing too was reversible, following the proposal by Shieber (1988), and implemented as a Uniform Tabular Algorithm (UTA), a data-driven selection function which was a generalization of the Earley deduction scheme. The UTA had a uniform indexing mechanism for items and an agenda-based control that allowed item sharing between parsing and generation: partial results computed in one direction could be computed in the other. Items would have either the LF or the string specified but not both: the parser would take items with instantiated string variables but with uninstantiated LFs, and vice-versa for the generator. This model therefore fulfilled some of the conditions required for CCs (reversibility and a degree of incrementality) but not all, as it was intended to parse its own utterances for on-line ambiguity checking (self-monitoring), rather than for interactivity and simultaneous interpretation of user input.

1.3.4 Incremental NLG for Dialogue

Recent work on incremental dialogue systems, driven by evidence that incremental systems are more efficient and pleasant to use than their non-incremental counterparts (Aist et al., 2007), has brought the challenges for interactive NLG to the fore. In particular, Schlangen and Skantze (2009, 2011)'s proposal for an abstract incremental architecture for dialogue, the *Incremental Unit* (IU) framework, has given rise to several interactive systems, including some with interesting NLG capabilities.

In Schlangen and Skantze's architecture, modules comprise a *left buffer* for input increments, a *processor*, and a *right buffer* for output increments. It is the *adding*, *commitment to* and *revoking* of IUs in a module's right buffer and the effect of doing so on another module's left buffer that determines system behaviour. Multiple competing IU hypotheses may be present in input or output buffers, and dependencies between them (*e.g.* the dependency of inferred semantic information from lexical information) are represented by *groundedIn*[4] relations between IUs.

The fact that all modules are defined in this way allows incremental behaviour throughout a dialogue system, and this has been exploited to create systems capable of some CC types, including the generation and interpretation of mid-utterance backchannels (Skantze and Schlangen, 2009) and interruptions (Buß et al., 2010). However, most of these systems have focussed on the incremental management of the dialogue, rather than on NLU and NLG themselves or on their interdependence. As a result, they tend to use canned text output for NLG;

[4] *groundedIn* links are transitive dependency relations between IUs specified by the system designer which may be exploited by modules. For instance, a word hypothesis IU may be grounded in a particular automatic speech recognition (ASR) result, and only added to the word hypothesizer's output graph once that part of the ASR graph is *committed*. See Skantze and Schlangen (2009) and Skantze and Hjalmarsson (2010) for more details.

consequently they lack interchangeability, and are therefore not suited for more complex CC phenomena.

Skantze and Hjalmarsson (2010), however, describe a model and system (Jindigo) which incorporates incremental NLG (although still using canned text rather than a more flexible approach). Jindigo can begin response generation before the end of a user utterance: as *word* hypotheses become available from incoming speech input, these are sent in real time to the NLU module, which in turn incrementally outputs *concept* hypotheses to the dialogue manager. This incrementally generates a *speech plan* for the speech synthesiser, which in turn can produce verbal output composed of speech *segments* divided into individual words. This incremental division allows Jindigo to begin speech output before speech plans are complete (*well, let's see ...*). It also provides a mechanism for self-repair in the face of changing speech plans during generation, when input concepts are revised or revoked. By cross-checking the speech plan currently being synthesised against the new speech plan, together with a record of the words so far output, the optimal word/unit position can be determined from which the repair can be integrated. Depending on the progress of the synthesiser through the current speech plan, this repair may be either *covert* (before synthesis) or *overt* (after synthesis), on both the segment and word levels. However, the use of different representations in NLU and NLG, together with the use of atomic semantic representations for entire multi-word segments in NLG, mean that our criteria of interchangeability and incremental semantic interpretation are not met, and a full treatment of CCs is still lacking.

1.3.5 Computational and Formal Approaches to Compound Contributions

Skantze and Hjalmarsson (2010) and Buß et al. (2010), as mentioned above, provide models that can handle some forms of compound contributions: mid- utterance backchannels, interruptions and (some) clarifications and confirmations. A few recent computational implementations and formal models focus specifically on more complex aspects of CCs.

DeVault et al. (2009, 2011) present a framework for predicting and suggesting completions for partial utterances: given partial speech recognition (ASR) hypotheses, their domain-specific classification-based approach can robustly predict the completion of an utterance begun by a user in real time. Given a corpus which pairs ASR output features from user utterances with the corresponding hand-annotated final semantic frames, they train a maximum entropy classifier to predict frames from a given partial ASR result. They achieve high precision in the correct selection of semantic frames, and provide some indication of possible transition points by using another classifier trained to estimate the point in the incoming utterance where the probability of the semantic frame currently selected being correct is unlikely to improve with further ASR results.

While their focus is on incremental interpretation rather than generation, this provides a practical model for part of the process involved in a CC: the jump from

partial NLU hypotheses to a suggested completion. DeVault et al. (2009, 2011) provide a basic NLG strategy for such completions by their system: by finding training utterances that match the predicted semantics against the partial input seen so far, the selection of the remainder of the utterance can be produced as the generator's completion. However, while such a model produces incremental semantic interpretations, its lack of syntactic information and its restriction to a finite set of semantic frames known in the domain prevent it from being a full model for CCs: such a model must be more flexible and able to account for syntactic constraints across speaker transitions.

Poesio and Rieser (2010), in contrast, describe a grammar-based approach that incorporates syntactic, semantic and pragmatic information via a lexicalised tree adjoining grammar (TAG) paired with the PTT model for incremental dialogue interpretation (Poesio and Traum, 1997). They provide a full account of the incremental interpretation process, incorporating lexical, syntactic and semantic information and meeting the criteria of incremental interpretation and representation. Beyond this, they also provide a detailed account of how a suggested collaborative completion might be derived using inferential processes and the recognition of plans at the utterance level: by matching the partial representation at speaker transition against a repository of known plans in the relevant domain, an agent can determine the components of these plans which have not yet been made explicit and make a plan to generate them. Importantly, the plans being recognised are at the level of speech planning: the desired continuation is determined by recognising the phrases and words observed so far as being part of a plan that makes sense in the domain and the current context.

This model therefore meets many of the criteria we defined: both interpretation and representation are incremental, with semantic and syntactic information being present; the use of PTT suggests that linguistic context can be incorporated suitably. However, while reversibility might be incorporated by the choice of suitable parsing and generation frameworks, this is not made explicit; and the extensibility of the representations seems limited by TAG's approach to adjunction (extension via syntactic adjuncts seems easy to treat in this approach, but more general extension is less clear). The use of TAG also seems to restrict the grammar to licensing grammatical strings, problematic for some CCs (see Section 1.2.2.2).

1.3.6 Summary

Previous work provides models for NLG that are incremental at the word-by-word level, and which can run in parallel with incremental parsing of user contributions, with some form of reversible representation. These models variously provide incremental syntactic construction during generation (Kempen and Hoenkamp, 1987, De Smedt, 1990) and incremental changing of the inputs to generation (Guhe, 2007, Skantze and Hjalmarsson, 2010). However, they do not generally explain how meaning is built up strictly incrementally – how partial

structures in generation can be related to maximal semantic content on a word-by-word basis. On the other hand, approaches specifically targeted at collaborative contributions and the required incremental modelling lack either strong incremental representation, so the parts of the utterance responsible for parts of the meaning representation cannot be determined (DeVault et al., 2009, 2011), or lack reversibility or extensibility while relying on licensing strings rather than meaning representations (Poesio and Rieser, 2010). In addition, little attention has been paid to the availability of linguistic context to NLG, and its sharing with NLU, on an incremental basis. An incremental approach is needed that not only has the qualities of reversibility and extensibility, but also the ability to generate incremental semantic interpretations and lexically anchored representations.

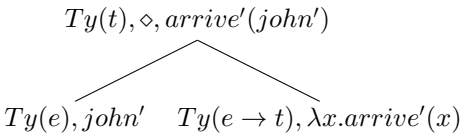
1.4 Dynamic Syntax (DS) and Type Theory with Records (TTR)

The approaches outlined so far all lack one or more of the criteria for a successful treatment of CCs. In this section, we describe an incremental grammar formalism and show how it can be extended to meet all these criteria, including strong incremental interpretation, incremental representation and reversibility.

1.4.1 Dynamic Syntax

One formalism with potential to satisfy the criteria for handling CCs described above is Dynamic Syntax (DS; (Kempson et al., 2001, Cann et al., 2005), *inter alia*). DS is an action-based and semantically-oriented incremental grammar formalism that dispenses with an independent level of syntax, instead expressing grammaticality via constraints on the word-by-word monotonic growth of semantic structures. In its original form, these structures are *trees*, with nodes corresponding to terms in the lambda calculus; these nodes are annotated with labels expressing their semantic type and formula, and beta-reduction determines the type and formula at a mother node from those at its daughters (1.21):

(1.21)

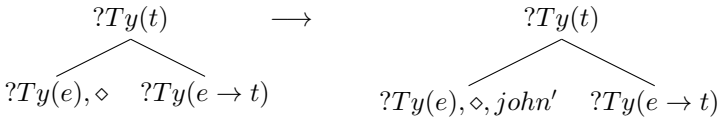


The DS lexicon comprises *lexical actions* associated with words, and also a set of globally applicable *computational actions*. Both of these are defined as monotonic tree update operations, and take the form of IF-THEN action structures. In traditional DS notation, the lexical action corresponding to the word *John* has the preconditions and update operations in (1.22). Trees are updated by these actions during the parsing process as words are consumed from the input string.

(1.22) *John* :

IF	$?Ty(e)$
THEN	$\texttt{put}(Ty(e))$
	$\texttt{put}(Fo(john'))$
ELSE	$\texttt{abort}$

(1.23)



DS parsing begins with an *axiom tree* (a single requirement for a truth value, $?Ty(t)$), and at any point, a tree can be *partial*, with nodes annotated with requirements for future development (written with a ? prefix) and a *pointer* (written $\diamond$) marking the node to be developed next (1.23). Actions can then satisfy and/or add requirements. Lexical actions generally satisfy a requirement for their semantic type, but may also add requirements for items expected to follow (*e.g.* a transitive verb may add a requirement for an object of type $Ty(e)$). Computational actions represent generally available strategies such as removing requirements which are already satisfied, and applying beta-reduction. (1.23) shows the application of the action for *John* defined in (1.22). Grammaticality of a word sequence is defined as satisfaction of all requirements (*tree completeness*) leading to a complete semantic formula of type $Ty(t)$ at the root node, thus situating grammaticality as *parseability*. The left-hand side of Figure 1.1 shows a sketch of a parse for the simple sentence *John likes Mary*: transitions represent the application of lexical actions together with some sequence of computational actions, monotonically constructing partial trees until a complete tree is yielded.

1.4.1.1 Generation by Parsing

Tactical generation in DS can be defined in terms of the parsing process and a subsumption check against a goal tree – a complete and fully specified DS tree such as (1.21) which represents the semantic formula to be expressed (Otsuka and Purver, 2003, Purver and Otsuka, 2003, Purver and Kempson, 2004a). The generation process uses exactly the same tree and action definitions as the parsing process, applied in the same way: trees are extended incrementally as words are added to the output string, and the process is constrained by checking for compatibility with the goal tree. Compatibility is defined in terms of tree *subsumption*: a tree subsumes a goal tree if it contains no nodes or node annotations absent in the goal tree.[5]

Generation thus follows a "parse-and-test" procedure: lexical actions are chosen from the lexicon and applied; and after each successful application, a subsumption check removes unsuitable candidates from the parse state – see the

[5] More correctly, if it contains no nodes which do not subsume some node in the goal tree; as node labels and addresses may be underspecified, the distinction is important for many syntactic phenomena, but we will ignore it here – see (Purver and Kempson, 2004a).

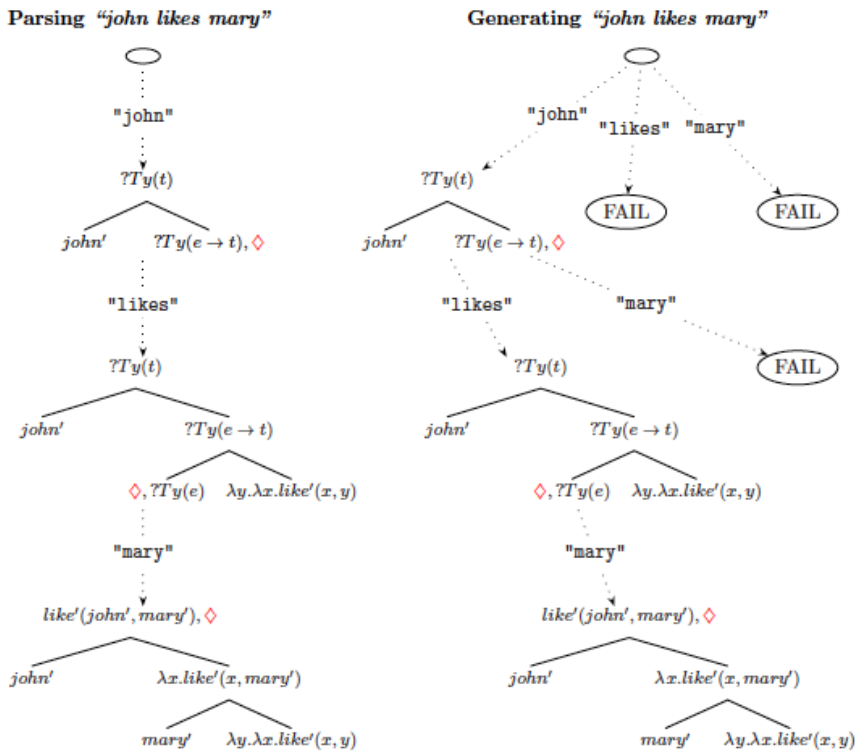


Figure 1.1 Parsing/generating *John likes Mary*, from (Purver and Kempson, 2004a)

right hand side of Figure 1.1 for a sketch of the process. From an NLG perspective, lexicalisation and linearisation (or in psycholinguistic terms, formulation and word ordering) are thus combined into one process: each word in the lexicon is tested for its applicability at each point of possible tree extension, and if accepted by the generator it is both selected and realised in the output string in one single action. As with Neumann (1998)'s framework, DS inherently has the quality of *reversibility*, as the input for generating a string is the semantic tree that would be derived from parsing that string.

1.4.1.2 Context in Dynamic Syntax

Access to some model of linguistic context is required for processing discourse phenomena such as anaphora and ellipsis. For DS, being an incremental framework, this context is taken to include not only the end product of parsing or generating a sentence (the semantic tree and corresponding string), but information about the dynamics of the parsing process itself – the lexical and computational action sequence used to build the tree. Strict readings of anaphora and verb phrase ellipsis (VPE) are obtained by copying semantic formulae (as node annotations) from context: anaphoric elements such as pronouns and elliptical auxiliaries annotate trees with typed metavariables, and computational rules allow

the substitution of a contextual value to update those metavariables. Sloppy readings are obtained by re-running a sequence of actions from context: a previous action sequence triggered by a suitable semantic type requirement (and resulting in a formula of that type) can be re-used, again providing a complete semantic formula for a node temporarily annotated with just a metavariable – see (Purver et al., 2006, Kempson et al., 2011) for details.

As defined in (Purver and Kempson, 2004b, Purver et al., 2006), one possible model for such a context can be expressed in terms of triples $\langle T, W, A\rangle$ of a tree T, a word-sequence W and the sequence of actions A, both lexical and computational, that are employed to construct the tree. In parsing, the parser state P at any point is characterised as a set of these triples; in generation, the generator state G consists of a goal tree T_G and a set of possible parser states paired with their hypothesised partial strings S. This definition of a generator state in terms of parse states ensures equal access to context for NLU and NLG, as required, with each able to use dynamically a full representation of the linguistic context produced so far.

1.4.1.3 Suitability for Compound Contributions

In the basic definitions of the formalism, DS fulfils some of our criteria to provide a model of CCs. It is inherently *incremental*: as each word is parsed or generated, it monotonically updates a set of partial trees. The definition of generation in terms of parsing means that it is also ensured to be *reversible*, with representations being naturally interchangeable between parsing and generation processes at any point. And context is defined to be incrementally and equally available to both parsing and generation.

Correspondingly, Purver and Kempson (2004b) outline a DS model for CCs, showing how the shift from NLU (hearer) to NLG (speaker) can be achieved at any point in a sentence, with context and grammatical constraints transferred seamlessly. The parser state at transition P_T (a set of triples $\langle T, W, A\rangle$) serves as the starting point for the generator state G_T , which becomes $(T_G, \{S_T, P_T\})$, where S_T is the partial string heard so far and T_G is whatever goal tree the generating agent has derived. The standard generation process can then begin, testing lexical and computational actions on the tree under construction in P_T , and successful applications will result in words extending S_T such that the tree is extended while subsuming T_G. The transition from speaker to hearer can be modelled in a directly parallel way, without the need to produce a goal tree, due to the interchangeability of generation and parse states and the context they contain. Gargett et al. (2009) also show how such a model can handle mid-sentence clarification and confirmation.

However, it is not clear that the DS model on its own fulfils all the conditions set out in Section 1.2.6. It produces representations that express semantic, syntactic and lexical information on an incremental basis; but in order to fulfil our criterion of strong incremental interpretation we require these to be in a form suitable for reasoning about possible meanings and continuations, and for

determining the contribution of words and phrases. It is not clear how an agent can reason from a partial semantic tree (without a semantic formula annotating its top node) to a goal for generation, especially if this goal must itself be in the form of a tree – and Purver and Kempson (2004a) give no account of how T_G can be derived. The account therefore lacks a way to account for how appropriate completions can be decided, and remains entirely tactical rather than strategic. Given this, there is also a question about the criterion of *extensibility*: while partial DS tree structures are certainly extendable, the lack of a clearly defined semantic interpretation at each stage means it is unclear whether extensibility applies in a semantic sense.

The criterion for incremental representation is also not entirely met. The model of context includes the contributions of the words and phrases seen, as it includes lexical and computational action sequences; but it is not clear how to retrieve from context the correspondence between a word and its contribution (information needed to resolve anaphora and clarifications). While word-action-formula correspondences are present for individual parse path hypotheses (individual $\langle T, W, A\rangle$ triples), there is no straightforward way to retrieve all action or formula hypotheses for a given word when the context contains a set of such triples with no explicit links between them.

1.4.2 Meeting the Criteria

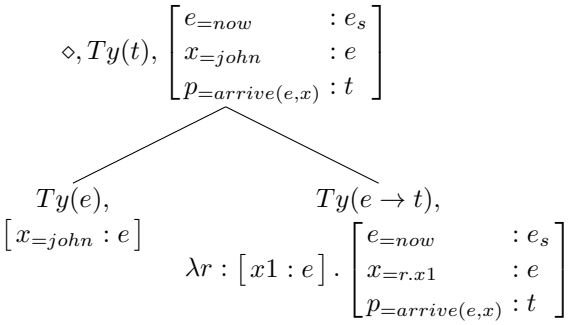
However, recent extensions to DS do provide a model that fulfils these missing criteria. The use of Type Theory with Records (TTR) for semantic representation permits incremental interpretation and full extensibility; and the use of a graph-based model of context permits incremental representation.

1.4.2.1 Type Theory with Records

Recent work in DS has started to explore the use of Type Theory with Records (TTR; (Betarte and Tasistro, 1998, Cooper, 2005, Ginzburg, 2012)) to extend the DS formalism, replacing the atomic semantic type and epsilon calculus formula node labels with more complex *record types*, and thus providing a more structured semantic representation. Purver et al. (2010) provide a sketch of one way to achieve this and show how it can be used to incorporate pragmatic information such as illocutionary force and participant reference (thus, amongst other things, giving a full account of the grammaticality of examples such as (1.13). Purver et al. (2011) introduce a slightly different variant using a Davidsonian event-based representation, and this is shown in (1.24) below. The semantic formula annotation of a node is now a TTR record type: a sequence of fields consisting of pairs of *labels* with *types*, written $[x : e]$ for a label x of type e. The identity of predicates and arguments is expressed by use of *manifest* types written $[x_{=john} : e]$ where $john$ is a singleton subtype of e. The standard DS type label $Ty()$ is now taken to refer to the final (*i.e.* lowest) field of the corresponding record type. Tree representations otherwise remain as before, with functional application of

functor nodes to argument nodes giving the overall TTR record type of the mother node.

(1.24) *John arrives* ↦



As well as providing the structure needed for representation of pragmatic information, the use of TTR allows us to provide a semantic representation at the root node of a tree, with this becoming more fully specified (via TTR *sub typing*) as more information becomes available. In TTR, a record type x is a subtype of a record type y if x contains at least all fields present in y, modulo renaming of labels, with x possibly also containing other fields not present in y. As Hough (2011) shows, this allows a version of DS in which root nodes are annotated with the maximal semantic content currently inferable given the labels present at the daughters; as more words are parsed (or generated) and the daughter nodes become more fully specified, the root node content is updated to a subtype of its previous type.

As Figure 1.2 shows, this provides a semantic representation (the TTR representation at the root node) that is incrementally updated to show the maximal semantic information known – precisely meeting our criterion of strong incremental *interpretation*. After the word *John*, we have information about an entity of manifest type *john* (a subtype of e), and know there will be some overall sentential predicate of type t, but don't yet know anything about the predicate or the argument role that *john* plays in it. As more words are added, this information is specified and the TTR subtype becomes more specific. This information was of course already present in partial DS tree structures, but implicit; this approach allows it to be explicitly represented, as required in CC generation (see Section 1.4.2.2).

The use of TTR also permits semantic *extensibility*, giving a straightforward analysis of continuations as extensions of an existing semantic representation. Adding fields to a record type results in a more fully specified record type that is still a subtype of the original. There is no requirement that the extension be via a complete syntactic constituent (*e.g.* an adjunct), as the semantic representation is available fully incrementally.

1.4.2.2 Parsing and Generation Context as a Graph

A further modification provides the required incremental *representation*. Rather than seeing linguistic context as centered around a set of essentially unrelated

$$\begin{array}{ccccc}
\begin{bmatrix} x_{=john} : e \\ p : t \end{bmatrix} & \mapsto & \begin{bmatrix} e_{=now} : e_s \\ x_{=john} : e \\ p_{=arrive(e,x)} : t \end{bmatrix} & \mapsto & \\
\textit{John} & \mapsto & \textit{John arrives} & \mapsto &
\end{array}$$

$$\begin{array}{ccc}
\begin{bmatrix} e_{=now} : e_s \\ x1 : e \\ x_{=john} : e \\ p1_{=by(e,x1)} : t \\ p_{=arrive(e,x)} : t \end{bmatrix} & \mapsto & \begin{bmatrix} e_{=now} : e_s \\ x1_{=plane} : e \\ x_{=john} : e \\ p1_{=by(e,x1)} : t \\ p_{=arrive(e,x)} : t \end{bmatrix} \\
\textit{John arrives by} & \mapsto & \textit{John arrives by plane}
\end{array}$$

Figure 1.2 Incremental interpretation via TTR subtypes

action sequences, an alternative model is to characterise it as a Directed Acyclic Graph (DAG). Sato (2011) shows how a DAG with DS *actions* for edges and (partial) *trees* for nodes allows a compact model of the dynamic parsing process; and Purver et al. (2011) extend this to integrate it with a word hypothesis graph (or "word lattice") as obtained from a standard speech recogniser.

This results in a model of context as shown in Figure 1.3, a hierarchical model with DAGs at two levels. At the action level, the parse graph DAG (shown in the lower half of Figure 1.3 with solid edges and circular nodes) contains detailed information about the actions (both lexical and computational) used in the parsing or generation process: edges corresponding to these actions connect nodes representing the partial trees built by them, and a path through the DAG corresponds to the action sequence for any given tree. At the word level, the word hypothesis DAG (shown at the top of Figure 1.3 with dotted edges and rectangular nodes) connects the words to these action sequences: edges in this DAG correspond to words, and nodes correspond to sets of parse DAG nodes (and therefore sets of hypothesised trees). Note that many possible word hypotheses may be present for NLU in a spoken dialogue system, as multiple ASR hypotheses may be available; in NLG, many possible competing word candidates may be being considered at any point. In both cases, this can be represented by alternative word DAG edges.

For any partial tree, the context (the words, actions and preceding partial trees involved in producing it) is now available from the paths back to the root in the word and parse DAGs. Moreover, the sets of trees and actions associated with any word or word subsequence are now directly available as that part of the parse DAG spanned by the required word DAG edges. This, of course, means that the contribution of any word or phrase can be directly obtained, fulfilling the criterion of *incremental representation*. It also provides a compact and efficient representation for multiple competing hypotheses, compatible with DAG representations commonly used in interactive systems, including the incremental dialogue system Jindigo (Skantze and Hjalmarsson, 2010) (see Section 1.3.4

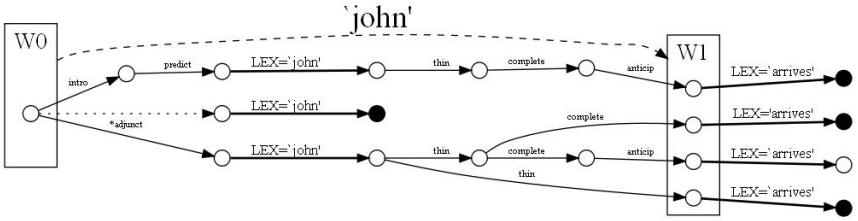


Figure 1.3 DS context as a DAG, consisting of a parse DAG (circular nodes=trees, solid edges=lexical(bold) and computational actions) *groundedIn* the corresponding word DAG (rectangular nodes=tree sets, dotted edges=word hypotheses) with word hypothesis *john* spanning tree sets W0 and W1

and below). Importantly, the DS definition of generation in terms of parsing still means this model will be equally available to both parsing and generation, and used in the same way by both. The criteria of *interchangeability* and equal access to *incremental context* are therefore still assured.

1.5 Generating Compound Contributions

Given this suitable framework for parsing and generation, we show how it can be used to provide a possible solution to the challenge posed by CCs for NLG process in dialogue, in line with the requirements described at the end of Section 1.2. We describe how an incremental dialogue system can handle the phenomenon through use of the incremental parsing and generation models of Dynamic Syntax (DS) combined with semantic construction of TTR record types.

1.5.1 The DyLan Dialogue System

DyLan[6] is a prototype incremental dialogue system based around Jindigo (Skantze and Hjalmarsson, 2010) and incorporating an implementation of DS-TTR to provide the NLU (Purver et al., 2011, Eshghi et al., 2011) and NLG (Hough, 2011) modules. Following Jindigo, it uses Schlangen and Skantze (2009)'s abstract model of incremental processing: each module is defined in terms of its *incremental unit* (IU) inputs and outputs. For the NLU module, input IUs are updates to a word hypothesis DAG, as produced by a speech recogniser; and output IUs are updates to the context DAG as described in Section 1.4.2.2 above, including the latest semantic representations as TTR record types annotating the root nodes of the latest hypothesised trees. For the NLG module, input IUs must be representations of the desired semantics (the goal for the current generation process), while output IUs are again updates to the

[6] DyLan stands for 'DYnamics of LANguage'.

context DAG, including the latest word sequence for output. The context DAG is shared by parser and generator: both modules have access to the same data structure, allowing both NLU and NLG processes to add incrementally to the trees and context currently under construction at any point.

1.5.1.1 Goal Concepts and Incremental TTR Construction

The original DS generation process required a semantic *goal tree* as input, but the strong incremental semantic interpretation property of the extended TTR model simplifies this requirement. As semantic interpretations (TTR record types) are available for any partial tree under construction, the generation process can now be driven by a *goal concept* in the form of a TTR record type, rather than a full goal tree. This reduces the complexity of the input, making the system more compatible with standard information state update and issue-based dialogue managers (*e.g.* (Larsson, 2002)). Goal concepts and output strings for the generator now take a form exactly like the partial strings and maximal semantic types for the parser shown in Figure 1.2.

The DyLan parsing system constructs a record type for each path-final tree in the parse DAG as each input word is received, allowing maximal semantic information for *partial* as well as complete trees to be calculated; this is implemented via a simple algorithm that places underspecified metavariables on nodes that lack TTR record types, and then continues with right corner-led beta reduction as for a complete tree (see (Hough, 2011) for details). As words in the lexicon are tested for generation, the generator checks for a *supertype* (subsumption) relation between path-final record types and the current goal record type, proceeding on a word-by-word basis. Parse paths that construct record types not in a supertype relation to the goal may be abandoned, and when a *type match* (*i.e.* subsumption in both directions) with the goal concept is found, generation is successful and the process can halt.

1.5.2 Parsing and Generation Co-Constructing a Shared Data Structure

The use of TTR record types in NLG removes the need for grammar-specific parameters (a real need when creating goal trees) and means that little modification is required for an off-the-shelf dialogue manager to give the system a handle on CCs. Domain knowledge can also be expressed via a small ontology of domain-specific record types. Born out of a long-standing use of frames for generating stereotypical dialogues in given situations (Lehnert, 1978) the idea of conversation *genres* (Ginzburg, 2012) can be employed here: domain concepts can be assumed to be of a given conversational TTR record type, as in the simple travel domain example below in Figure 1.4; this can contain underspecified fields (the $x1, x2, x3$ values) for information that varies by user and context, as well as fully specified manifest fields.

The interchangeability of representations betwen NLU and NLG means that the construction of a data structure such as that in Figure 1.4 can become a

$$\begin{bmatrix} e & : e_s \\ x3 & : e \\ x2 & : e \\ x1 & : e \\ x_{=user} & : e \\ p3_{=by(e,x3)} & : t \\ p2_{=from(e,x2)} & : t \\ p1_{=to(e,x1)} & : t \\ p_{=go(e,x)} & : t \end{bmatrix}$$

Figure 1.4 A TTR record type representing a simple travel domain concept

collaborative process between dialogue participants, permitting a range of varied user input behaviour and flexible system responses. As with Purver et al. (2006)'s original model for CCs, the use of the same representations by NLU and NLG guarantees both the ability to begin parsing from the end-point of any generation process (even mid-utterance), and to begin generation from the end-point of any parsing process. Both NLU and NLG models are now characterised entirely by the parse context DAG, with the addition for generation of a TTR goal concept. The transition from generation to parsing now becomes almost trivial: the parsing process can continue from the final node(s) of the generation DAG, with parsing actions extending the trees available in the final node set as normal.

The transition from parsing to generation also requires no change of representation, with the DAG produced by parsing acting as the initial structure for generation (see Figure 1.5); now, though, we also require the addition of a goal concept to drive the generation process. But given the full incremental interpretation provided by the use of record types throughout, we can now also see how a generator might produce such a goal at a speaker transition.

1.5.3 Speaker Transition Points

The same record types are now used throughout the system: as the concepts for generating system plans, as the goal concepts in NLG, and for matching user input against known concepts in suggesting continuations. In interpretation mode, the emerging conversational record type in context can be incrementally checked against any known domain concept record types; if generation of a continuation is now required, this can be used to select the next generation goal on the basis of any matching knowledge in the system's knowledge base or information state. A suitable generation goal can be any subtype of the current top record type in the parse DAG at speaker transition; a match against a known concept (from domain or conversational context) can provide this.

Given the formal tools of DS-TTR parsing and generation to license CCs, we can therefore equip a system with the ability to generate them quite simply. Possible system transition points trigger the alternation between modules in their co-

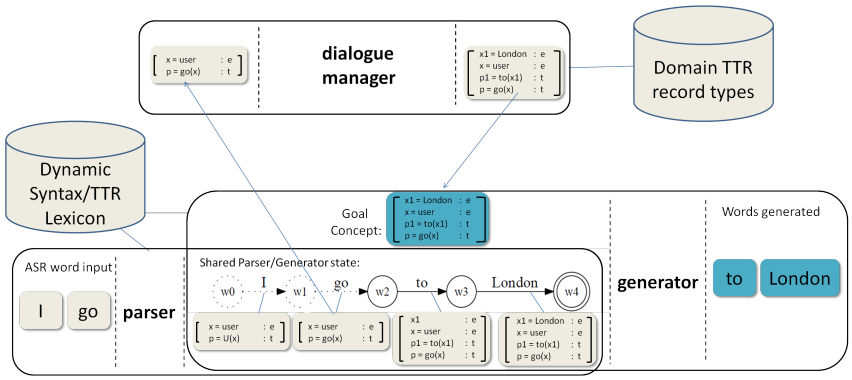


Figure 1.5 Completion of a compound contribution using incremental DS-TTR record type construction, with parser and generator constructing a shared parse state

construction of the shared parse/generator DAG; in DyLan, this is provided by a simplistic dialogue manager with high-level methods without reference to syntax or lexical semantics. We may employ a single method, `continueContribution` which simply reacts to a silence threshold message from the ASR module, halts the parser and selects a appropriate goal TTR record type from its concept ontology – achieving this by searching for the first concept that stands in a subtype relation to the record type under construction. The selected record type then acts as the new goal concept, allowing generation to begin immediately. While speech act information can be delivered to the synthesiser for the purposes of prosody alteration, in terms of semantics, no additional information about the utterance is required for a continuation. This stands in contrast to Poesio and Rieser (2010)'s account, which requires inference about speaker intentions; of course, this is not to say that such processes might not be useful, or required for certain situations to fully capture the dialogue data. Importantly, though, we can provide a model for the underlying mechanisms, and for the suggestion of simple continuations on the basis of domain knowledge, without such inference.

To ensure coherence between the different utterances making up a CC, the stipulation that a goal record type selected upon user silence be a subtype of the record type under construction facilitates the joint construction of a completed record type (see Figure 1.5). However, this does not mean the system must have a complete domain concept as a goal, as the selected goal may be an underspecified supertype of a domain concept. This allows contributions even when the system does not possess full information, but knows something about the continuation required, as in (1.25). Here, generation can begin at the first speaker transition if a suitable goal concept can be obtained (*e.g.* as in Figure 1.4, giving the information that a mode of transport is required but not the reference itself) – the word *from* can therefore be generated. At this point generation will then stop as the utterance covers all information in this goal concept (*i.e.* leaving no

un-vocalized goal information left to drive further generation), at which point parsing can take over again if user input is available:

(1.25) USER: *I'm going to London*
SYS: *from ...*
USER: *Paris.*

If the user interrupts or extends a system utterance, Jindigo's continuously updating buffers allow notification of input from the ASR module to be sent to the dialogue manager quickly[7] and the switch to parsing mode may take place. The dialogue manager's method `haltGeneration` stops the NLG and transfers the parse DAG construction role back to the NLU. Upon the generation or recognition of each word, a *strong incremental representation* can be extracted for the utterance so far, as each word is parsed semantically and syntactically, with incremental self-monitoring (Levelt, 1989) coming for free in that the utterance's string does not need to be passed back to a parsing module. This is not possible in string-based approaches (*e.g.* (DeVault et al., 2009)).

Extension contributions (*e.g.* adjuncts or prepositional phrases in the limited travel domain below, such as *to Paris* and *on Monday*, but in principle any extension) can be dealt with straightforwardly in both parsing and generation, as they introduce subtypes of the record type under construction. The user over-answering a prompt to extend a CC as in (1.26) is therefore handled straightforwardly, as the goal concept during generation may be overridden by the user's input if it is commensurate with the record type constructed up to the speaker transition (*i.e.* stands in a subtype relation to it). In this sense, a continuation from the user can be "unexpected" but not destructive to the continual build up of meaning, or the maintenance of the parse DAG.

(1.26) USER: *I want to go ...*
SYS: *to ...*
USER: *Paris from London*

The system is therefore capable of taking part in arbitrary speaker transitions, including multiple transitions during one co-constructed utterance, and generating any part of a contribution whose parse path will lead to constructing a domain concept record type. At any word position, and *a fortiori*, at any syntactic position, in the utterance, the module responsible for building this path may change depending on the user's behaviour, consistent with the psycholinguistic observations summarised in Section 1.2.

[7] A demonstration of the system's capability for rapid turn-taking is shown in (Skantze and Schlangen, 2009).

1.6 Conclusions and Implications for NLG Systems

In this chapter, we have outlined the phenomenon of compound contributions (CCs), detailed some of the many forms they can take, and explained why they are of interest to NLG researchers. CCs provide a stringent set of criteria for NLG itself and for NLG/NLU interaction. As set out in Section 1.2.6, these criteria entail full word-by-word incrementality in terms of representation, interpretation and context access, while requiring full interchangeability of representations between NLU and NLG. While previous research has produced NLG and dialogue models that provide incrementality in many ways, none of them fulfils all of these criteria.

In particular, we have seen that the use of standard string-licensing grammars is problematic: contextual pronominal reference changes with speaker transitions, resulting in successful, acceptable utterances which would have to be characterised as ungrammatical if considered merely as surface strings. We have also seen that neither lexical, syntactic nor semantic incremental processing is sufficient on its own; a fully CC-capable system must produce incremental representations of meaning, structure and lexical content *together*. However, by extending Dynamic Syntax to incorporate a structured, type-theoretic semantic representation and a graph-based context model, we can provide a model which meets all the criteria for handling CCs, and use this within a prototype interactive system.

Speaker Intentions
One feature of note is that our framework allows us to model CCs without *necessarily* relying on speaker intention recognition. While intention recognition may well play a role in many CC cases, and may be a strategy available to hearers in many situations, our model does not rely on it as primary, instead allowing parsing and generation of CCs based only on an agent's internal knowledge and context. Such a model is compatible with existing approaches to interactive systems based on *e.g.* information state update rather than higher-order reasoning about one's interlocutor.

Alignment
The main focus of this chapter has been on providing a model that can license the grammatical features of the CC phenomena in question: one which is capable of generating (and parsing) the phenomena in principle. However, NLG systems in real interactive settings need to look beyond this, to features that characterise the naturalness or human-like qualities of the discourse and give us a way to choose between possible alternative formulations. One such feature is *alignment* – the tendency of human interlocutors to produce similar words and/or structures to each other (Pickering and Garrod, 2004). Giving an full account of alignment in the DS-TTR framework is a matter for future research (for one thing, requiring a general model of preferences in DS parsing dynamics – for an initial model see

(Sato, 2011)), and we see this as an area of interest for NLG research. We note here, though, that the graphical model of context does provide an interesting basis for such research. Taking the context DAG as a basis for lexical action choice – with a basic strategy being to re-use actions in context by DAG search before searching through the NLG lexicon – provides an initial platform for an explanation of alignment. More recently used words would tend to be re-used, and the sharing and co-construction of the context model between parsing and generation explains how this happens between hearing and speaking (and vice versa). Interestingly, however, this model would predict that syntactic alignment should arise mainly from lexical alignment – through re-use of lexical action sequences – rather than being driven as an individual process. Recent empirical data suggest that this may indeed be so, with syntactic alignment in corpora perhaps explainable as due to lexical repetition (Healey et al., 2010).

Coordination and Repair
It is also worth noting briefly here that speaker transition in CCs is often associated not merely with a smooth transition, but with self-repair phenomena such as repetition and reformulation, as well as the other-repair phenomena such as mid-utterance clarification we have already described. A full model must account for this, and we look forward to NLG research that incorporates self- and other-repair together into an incremental model of speaker change and CCs. One proposal for how to go about this within the framework described here, using backtracking along the context graph to model repetition and reformulation, is currently being investigated (Hough, 2011).

Acknowledgments

This research was carried out under the *Dynamics of Conversational Dialogue* project, funded by the UK ESRC (RES-062-23-0962), with additional support through the *Robust Incremental Semantic Resources for Dialogue* project, funded by the EPSRC (EP/J010383/1). We also thank Ruth Kempson, Chris Howes, Arash Eshghi, Pat Healey, Wilfried Meyer-Viol and Graham White for many useful discussions.

References

Aist, G., Allen, J. F., Campana, E., Gomez Gallo, C., Stoness, S., Swift, M., and Tanenhaus, M. K. (2007). Incremental dialogue system faster than, preferred to its nonincremental counterpart. In *Proceedings of the Annual Conference of the Cognitive Science Society (henceforth 'CogSci')* .

Allen, J. F., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001). Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–37.

Asher, N. and Lascarides, A. (2008). Commitments, beliefs and intentions in dialogue. In *Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue (henceforth 'SEMDIAL')*.

Betarte, G. and Tasistro, A. (1998). Extension of Martin-Löf type theory with record types and subtyping. In *25 Years of Constructive Type Theory*, pages 21–40. Oxford University Press, Oxford, UK.

Brennan, S. E. and Schober, M. (2001). How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44(2):274–296.

Burnard, L. (2000). Reference guide for the british national corpus (world edition). Available from: http://www.natcorp.ox.ac.uk/archive/worldURG/urg.pdf.

Buß, O., Baumann, T., and Schlangen, D. (2010). Collaborating on utterances with a spoken dialogue system using an ISU-based approach to incremental dialogue management. In *Proceedings of the SIGdial Conference on Discourse and Dialogue (henceforth 'SIGDIAL')*.

Cann, R., Kaplan, T., and Kempson, R. (2005). Data at the grammar-pragmatics interface: the case of resumptive pronouns in English. *Lingua*, 115(11):1551–1577.

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

De Smedt, K. (1990). IPF: An incremental parallel formulator. In *Current research in natural language generation*, pages 167–192. Academic Press, San Diego, CA.

De Smedt, K. (1991). Revisions during generation using non-destructive unification. In *Proceedings of the European Workshop on Natural Language Generation (henceforth 'ENLG')*.

De Smedt, K., Horacek, H., and Zock, M. (1996). Some problems with current architectures in natural language generation. In Adorni, G. and Zock, M., editors, *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pages 17–46. Springer LNCS, Berlin, Germany.

DeVault, D., Sagae, K., and Traum, D. (2009). Can i finish?: learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of SIGDIAL*.

DeVault, D., Sagae, K., and Traum, D. (2011). Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue and Discourse*, 2(1):143–170.

Eshghi, A., Purver, M., and Hough, J. (2011). DyLan: Parser for dynamic syntax. Technical Report EECSRR-11-05, School of Electronic Engineering, Computer Science, Queen Mary University of London.

Ferreira, V. S. (1996). Is it better to give than to donate? syntactic flexibility in language production. *Journal of Memory and Language*, 35(5):724–755.

Gargett, A., Gregoromichelaki, E., Kempson, R., Purver, M., and Sato, Y. (2009). Grammar resources for modelling dialogue dynamically. *Cognitive Neurodynamics*, 3(4):347–363.

Ginzburg, J. (2012). *The Interactive Stance: Meaning for Conversations*. Oxford University Press, Oxford, UK.

Goodwin, C. (1979). The interactive construction of a sentence in natural conversation. In Psathas, G., editor, *Everyday Language: Studies in Ethnomethodology*, pages 97–121. Irvington, New York, NY.

Goodwin, C. (1981). *Conversational organization: Interaction between speakers and hearers*. Academic Press, New York, NY.

Gregoromichelaki, E., Kempson, R., Purver, M., Mills, G. J., Cann, R., Meyer-Viol, W., and Healey, P. G. (2011). Incrementality and intention-recognition in utterance processing. *Dialogue and Discourse*, 2(1):199–233.

Guhe, M. (2007). *Incremental conceptualization for language production*. Lawrence Erlbaum Associates, Mahwah, NJ.

Guhe, M., Habel, C., and Tappe, H. (2000). Incremental event conceptualization and natural language generation in monitoring environments. In *Proceedings of the International Natural Language Generation Conference (henceforth 'INLG')*.

Healey, P. G. (2008). Interactive misalignment: The role of repair in the development of group sub-languages. In Cooper, R. and Kempson, R., editors, *Language in Flux: Dialogue Coordination, Language Variation, Change and Evolution*. College Publications, London, UK.

Healey, P. G., Purver, M., and Howes, C. (2010). Structural divergence in dialogue. In *Proceedings of the Conference on Architectures and Mechanisms for Language Processing*.

Hough, J. (2011). Incremental semantics driven natural language generation with self-repairing capability. In *Proceedings of the International Conference*

on Recent Advances in Natural Language Processing (henceforth 'RANLP').

Howes, C., Purver, M., Healey, P. G., Mills, G. J., and Gregoromichelaki, E. (2011). On incrementality in dialogue: Evidence from compound contributions. *Dialogue and Discourse*, 2(1):279–311.

Joshi, A. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty, D., Karttunen, L., and Zwicky, A. M., editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 206–250. Cambridge University Press, Cambridge, UK.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kay, M. (1985). Parsing in functional unification grammar. In Dowty, D., Karttunen, L., and Zwicky, A. M., editors, *Natural language parsing: psychological, computational and theoretical perspectives*, pages 251–278. Cambridge University Press, Cambridge, UK.

Kempen, G. and Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science*, 11(2):201–258.

Kempson, R., Cann, R., Eshghi, A., Gregoromichelaki, E., and Purver, M. (2011). Ellipsis. In Lappin, S. and Fox, C., editors, *The Handbook of Contemporary Semantic Theory.* Wiley, New York, NY, 2nd edition.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic Syntax: The Flow of Language Understanding.* Blackwell, Oxford, UK.

Larsson, S. (2002). *Issue-based Dialogue Management.* PhD thesis, Department of Linguistics, Göteborg University.

Lascarides, A. and Asher, N. (2009). Agreement, disputes and commitments in dialogue. *Journal of Semantics*, 26(2):109–158.

Lehnert, W. G. (1978). *The Process of Question Answering: A Computer Simulation of Cognition.* Lawrence Erlbaum Associates.

Lerner, G. H. (1991). On the syntax of sentences-in-progress. *Language in Society*, 20(3):441–458.

Lerner, G. H. (1996). On the "semi-permeable" character of grammatical units in conversation: Conditional entry into the turn space of another speaker. In Ochs, E., Schegloff, E. A., and Thompson, S. A., editors, *Interaction and Grammar*, pages 238–276. Cambridge University Press, Cambridge, UK.

Lerner, G. H. (2004). Collaborative turn sequences. In Lerner, G. H., editor, *Conversation Analysis: Studies from the First Generation*, pages 225–256. John Benjamins, Amsterdam, The Netherlands.

Levelt, W. J. M. (1983). Monitoring and self-repair in speech. *Cognition*, 14(1):41–104.

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation.* MIT Press, Cambridge, MA.

Milward, D. (1991). *Axiomatic Grammar, Non-Constituent Coordination, Incremental Interpretation*. PhD thesis, University of Cambridge.

Neumann, G. (1994). *A Uniform Computational Model for Natural Language Parsing and Generation*. PhD thesis, Universitaat des Saarlandes, Saarbrücken.

Neumann, G. (1998). Interleaving natural language parsing and generation through uniform processing. *Artificial Intelligence*, 99(1):121–163.

Neumann, G. and van Noord, G. (1994). Reversibility and self-monitoring in natural language generation. In Strzalkowski, T., editor, *Reversible Grammar in Natural Language Processing*, pages 59–96. Kluwer, Dordrecht, The Netherlands.

Ono, T. and Thompson, S. A. (1993). What can conversation tell us about syntax? In Davis, P., editor, *Alternative Linguistics: Descriptive and Theoretical Modes*, pages 213–271. John Benjamins, Amsterdam, The Netherlands.

Otsuka, M. and Purver, M. (2003). Incremental generation by incremental parsing: Tactical generation in dynamic syntax. In *Proceedings of the CLUK Colloquium*.

Pickering, M. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–226.

Poesio, M. and Rieser, H. (2010). Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1:1–89.

Poesio, M. and Traum, D. R. (1997). Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

Purver, M., Cann, R., and Kempson, R. (2006). Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2–3):289–326.

Purver, M., Eshghi, A., and Hough, J. (2011). Incremental semantic construction in a dialogue system. In *Proceedings of the International Conference on Computational Semantics*.

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., and Cann, R. (2010). Splitting the 'I's and crossing the 'you's: Context, speech acts and grammar. In *Proceedings of SEMDIAL*.

Purver, M., Howes, C., Healey, P. G., and Gregoromichelaki, E. (2009). Split utterances in dialogue: a corpus study. In *Proceedings of SIGDIAL*.

Purver, M. and Kempson, R. (2004a). Context-based incremental generation for dialogue. In *Proceedings of INLG*.

Purver, M. and Kempson, R. (2004b). Incrementality, alignment and shared utterances. In *Proceedings of SEMDIAL*.

Purver, M. and Otsuka, M. (2003). Incremental generation by incremental parsing: Tactical generation in dynamic syntax. In *Proceedings of ENLG*.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.

Rühlemann, C. and McCarthy, M. (2007). *Conversation in Context: A Corpus-Driven Approach*. Continuum, London, UK.

Sato, Y. (2011). Local ambiguity, search strategies and parsing in dynamic syntax. In Gregoromichelaki, E., Kempson, R., and Howes, C., editors, *The Dynamics of Lexical Interfaces*. CSLI Publications, Stanford, CA.

Saxton, M. (1997). The contrast theory of negative input. *Journal of Child Language*, 24(1):139–161.

Schegloff, E. (1979). The relevance of repair to syntax-for-conversation. In Givon, T., editor, *Discourse and Syntax*, pages 261–286. Academic Press, New York, NY.

Schegloff, E. (2007). *Sequence organization in interaction: A primer in conversation analysis I*. Cambridge University Press, Cambridge, UK.

Schlangen, D. and Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics (henceforth 'EACL')*.

Schlangen, D. and Skantze, G. (2011). A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111.

Shieber, S. M. (1988). A uniform architecture for parsing and generation. In *Proceedings of the International Conference on Computational Linguistics (henceforth 'COLING')*.

Skantze, G. and Hjalmarsson, A. (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of SIGDIAL*.

Skantze, G. and Schlangen, D. (2009). Incremental dialogue processing in a micro-domain. In *Proceedings of EACL*.

Skuplik, K. (1999). Satzkooperationen. definition und empirische untersuchung. Technical Report 1999/03, Bielefeld University.

Sturt, P. and Crocker, M. (1996). Monotonic syntactic processing: a cross-linguistic study of attachment and reanalysis. *Language and Cognitive Processes*, 11(5):449–494.

Szczepek, B. (2000). Formal aspects of collaborative productions in English conversation. *Interaction and Linguistic Structures*, (17).

Thompson, H. (1977). Strategy and tactics: A model for language production. In *Papers from the Regional Meeting of the Chicago Linguistic Society*.

van Wijk, C. and Kempen, G. (1987). A dual system for producing self-repairs in spontaneous speech: Evidence from experimentally elicited corrections. *Cognitive Psychology*, 19(4):403–440.