Dynamic Intention Structures for Dialogue Processing

Charles L. Ortiz, Jr.
Natural Language and AI Laboratory
Nuance Communications
Sunnyvale, CA U.S.A.
`charles.ortiz@nuance.com`

Jiaying Shen
Natural Language and AI Laboratory
Nuance Communications
Sunnyvale, CA U.S.A.
`jiaying.shen@nuance.com`

Abstract

We examine personal assistance dialogues and argue that some form of constraint relaxation is necessary during dialogue processing as often only a subset of the constraints present in the intentional structure reflecting earlier parts of a dialogue can be satisfied in the context of a new utterance. We combine a fine-grained formal representation of intention with a non-monotonic consistency-based intention revision process to support a model of structured and evolving propositional content that leads to a flexible discourse segmentation process. The approach provides a bridge between models of rational agency, plan-based models of dialogue and theories of dynamic discourse semantics.

1 The problem

Plan-based approaches to dialogue processing structure a dialogue hierarchically into discourse segment purposes (roughly, intentions) and their interrelation (Grosz and Sidner, 1986). Such structures are incrementally elaborated as a dialogue unfolds and are referred to as *intentional structures*. They are grounded in agent models of collaboration and formal models of intention. The interpretation of a new utterance is given relative to whether it signals the start of a new segment, contributes to an existing one or completes it. The notion of contribution to a segment is given in terms of whether the sub-task (or its constraints) reflected by the utterance can play a part in the success of the task corresponding to the embedding segment. So, for example, in the task dialogues between experts and apprentices discussed in Grosz and Sidner (Grosz and Sidner, 1986; Lochbaum, 1998) an expert can suggest an action to an apprentice who will either execute it, ask for clarification or report obstacles.

Dialogues between a VPA and a user, however, differ in an important way from expert-apprentice task-based dialogues: a user typically provides some initial constraints on a task that it seeks help on after which the VPA will attempt to formulate a plan to satisfy those constraints. Often, however, either only a subset of those constraints can be satisfied or the user might change his mind on the set of constraints during the dialogue. Consequently, it may not be possible to accommodate a new utterance during the discourse segmentation process if one does so in terms of whether the interpretation of the utterance is consistent with the constraints so far articulated in the current segment. Rather, the system must relax some constraints to properly situate the utterance within the existing discourse structure and then continuing with the dialogue or providing assistance.

Consider the following example of a possible dialogue between a person and a virtual personal assistant (VPA) of the future in the context of a request for help in organizing a meeting with some friends after a conference session.

1. [User:] I want to plan a get together after the last session.
2. [System:] At what time?
3. [User:] 7pm.
4. [System:] OK.
5. [User:] Book a table at an Italian restaurant near the hotel and let Brian know.
6. [System]: Zingari is available at 7pm.
7. [User:] That's good.

Consider the following possible alternative user continuations in the highlighted contexts.

8a. And I'd like to include some good wine. (*Zingari does not have a good wine list. An alternative, Barbacco, does but it is farther away.*)

8b. Reserve a table at Chevy's instead. (*Chevy's is a Mexican restaurant.*)

8c. I decided that I want Spanish food.

8d. Actually, let's just go to a place for drinks. (*Zingari is not available for just drinks.*)

Consider each of these in the context of a SharedPlans (Grosz and Kraus, 1996) plan augmentation algorithm (Lochbaum, 1998). Lochbaum's algorithm determines the contribution of the interpretation of the current utterance, u, through a construction process which builds a complex recipe structure (action decomposition hierarchy) from simpler two-level recipes. In the last step, the algorithm checks for consistency: if the new constraints from u are satisfiable with those so far articulated in the recipe structure then they are combined, otherwise the algorithm fails and the user is alerted or queried.

Returning to the above examples, in each of the continuations the user introduces new constraints into the planning process[1] As it happens, each new constraint is inconsistent with the constraints communicated so far in the dialogue. In (8a), the system must engage in some constraint relaxation as otherwise the segment initiated by (5) would fail (and, hence, (8a) would have to be interpreted as part of the higher level segment: roughly, "I want to plan a get together with Brian and include some good wine"). This is to be expected and constitutes the reason the user needs assistance in the first place: he has no idea whether Zingari has a good wine list. The system may then try to find something a little farther away that meets all of the other constraints. In continuation (8b), one cannot simply delete the identity of the restaurant of Zingari and substitute that of Chevy's: there is also an inconsistency with a side-effect of the choice of Chevy's: the fact that it is a *Mexican* restaurant. In case (8c), the constraint that the restaurant be Italian is retracted which entails that the Zingari reservation be withdrawn. In case (8d), the method of setting up a get together is changed: the user decides to have drinks (nominally, at a bar) instead of going to a restaurant. However, a search should then not be constrained to an *Italian* bar. In all of these cases the binding for the objects in the second part of the action ("let Brian know") must depend on the different choices from the individual cases (so that if Zingari is picked, the VPA informs Brian and if the choice is changed to Barbacco the VPA informs Brian of the new location).

There are other concerns that these examples raise. Intention revision must be able to modify propositional content in a fine-grained way, rather than just deleting an inconsistent intention. For example, if we consider the first conjunct of (5), we would have something like:

$$\begin{aligned}&intends(System, \exists x \exists t.occurs(book(x), t)\\ &\quad \wedge table(x) \wedge restaurant(x) \wedge italian(x))\end{aligned}$$

Notice the embedded existential quantifier: the system has not fixed the identity of the restaurant or the time of the booking action. By utterance (8), however, those decisions have been made and the content within the scope of the above modal intention operator must be accessed and updated, without having to re-write the entire formula or delete the entire formula if the revising component is inconsistent with the intention: one would like to minimally modify the *contents* of the intention, unlike that in the belief revision literature. Dynamic Intention Structures (DIS), developed for modeling rational agents (Ortiz and Hunsberger, 2013; Hunsberger and Ortiz, 2008), addressed these problems. Whereas DRT makes use of a dynamic logic to deal with dynamic scoping of quantifiers, the theory of DIS's extends that idea to modalities with hierarchically structured content: the structure informs the consistency based revision procedure which lumps related elements, allowing incremental revision.

The similarity to DRT also addresses a perceived need, that has has been pointed out by others (Asher and Lascarides, 2003), to create a bridge between plan-based approaches and discourse *semantics* in a manner similar to approaches grounded in DRT.

2 Dynamic Intention Structures

We will present different forms of DISs that each serve different purposes. A *canonical* DIS is of the form $\langle V_c, T_c, int[\langle V_t, T_t, \langle\langle Id_r, A_r, V_r, T_r, E_r, C_r, S_r \rangle\rangle \rangle] \rangle$. V_c is a set of variables ("c" for "context") and T_c is a time point coinciding with the time of the intention. These external variables and time are existentially quantified in the translation to first order logic (FOL). V_t is a set of variables

---

[1] A reasonable segmentation would consist of a top-level segment for (1) and two sub-segments for (2)-(4) and (5)-(7). The DIS of Figure 1 lumps together the representations of (1)-(4). We do not delve into the precise mechanism behind the segmentation process as it does not bear directly on our presentation. A more detailed presentation would require a review of SharedPlans which are used to guide that process.

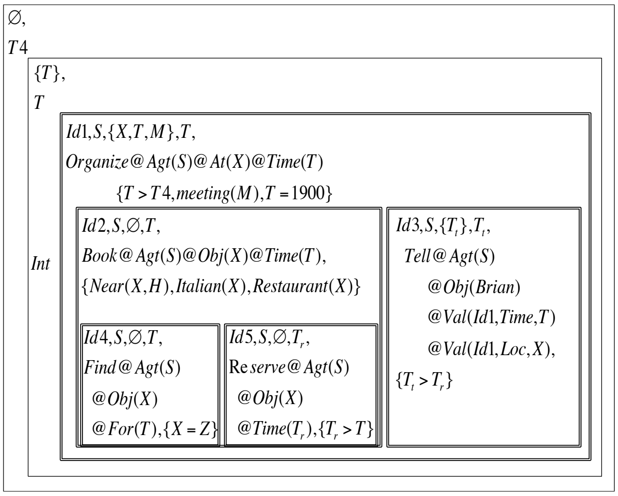


Figure 1: DIS after utterance 8.

and T_t is the time of the intended action We call each element within the double angle brackets ($\langle\!\langle \ldots \rangle\!\rangle$) a *plan tree node*. Each node has a unique identifier, Id_r ("r" for "root"); a possibly empty list of child nodes, S_r; an agent, A_r; an action, E_r; a set of local variables, V_r; and a set of constraints, C_r on actions and objects. Variables local to the intention are existentially quantified in the semantics, which allows one to express partially elaborated intentions such as "John intends to reserve *a room*" without necessarily fixing the identity of the room or the method for accomplishing the reservation.

The action representation makes use of an act-type constructor, @, which allows the construction of more complex act *types* from simpler ones (Ortiz, 1999). For example, suppose $drive@agt(A)@to(Boston)$ represents the act type of agent A driving to Boston. That act type could later be elaborated with further detail. For example, $drive@agt(A)@to(Boston)@on(Interstate95)$, might represent the act type of agent A driving to Boston via Interstate 95. In this way, the act-type constructor enables the representation of partially specified intentions without committing to a particular predicate arity—such as, $drive(Agent, Object)$.

To deal with partiality of action descriptions more systematically, the arguments to act-type modifiers will often be restricted to variables. For example, the preferred description of agent A driving to Boston would be $do(drive@agt(x)@to(y)) \land (x = A) \land (y = Boston)$. This technique has the advantage of enabling complex revisions to be performed simply by assigning or de-assigning values to variables.

DIS's, like DRSs, can be conveniently visualized using box notation. Figure 1 depicts the DIS obtained after utterances 1-8 (the constant h stands for the hotel). The representation is built up incrementally: the two outer boxes and the first box in the scope of the intention, labeled $Id1$, is a consequence of utterances (1-4). It says that at time $T4$,[2] the agent S (for system), intends to organize a meeting, M, at some yet to be specified location, X, (represented by the partially specified act-type term, $organize@Agt(S)@obj(M)@At(X)@Time(T)$, with constraint $meeting(M)$) at time $T = 1900$ (7 PM) in the future.

Utterances (5-7) lead to the full structure depicted in Figure 1: sub-actions corresponding to boxes $Id2$ and $Id3$ are introduced to capture the user's requirement that the meeting be organized by, respectively, booking a table at a nearby Italian restaurant $Book@Agt(S)@Obj(X)@Time(T)$ with constraints, $Near(X, H), Italian(X)$ and then telling Brian after it is reserved at time T_r (i.e., at time $T_t > T_r$). The system further decomposes this structure by adding sub-actions corresponding to $Id4$ and $Id5$: the choice of the restaurant Zingari (i.e., Z) is captured in the constraint $X = Z$ in $Id4$. Collectively, the DIS (minus the Id4 and Id5 boxes which have not been discussed and are planned in the background by the system) reflects the dialogue intentional structure.

3 Semantics of DISs

The semantics of any DIS in canonical form is specified by translating it into an FOL formula in a meta-language, $\mathcal{L}$. The translation of a DIS, $\mathcal{D}$, relative to a world, w, and an intention base, I, is written $\|\mathcal{D}\|_w^I$. The specification of the translation function makes use of a reification approach similar to that employed in the context of reasoning about knowledge (Moore 1985). The meta-language, $\mathcal{L}$, contains: (1) the usual logical connectives, $\{\land, \supset, \neg\}$, that stand for conjunction, implication and negation, respectively; (2) a set of meta-language constants that stand for variables and constants in the object language (i.e., the DIS language); and (3) a

---

[2] See Appendix for an explanation of the time index "T4".

1.$\|R(T_1,\ldots,T_n)\|_w^I = r(\|t_1\|_w^I \ldots, \|t_n\|_w^I)$, r a function
2. $\|T_1 = T_2\|_w^I = eq(t_1, t_2)$
3. $\|\langle V, C\rangle\|_w^I = exists(v, \|C\|_w^I)$,
4. $\|\neg\phi\|_w^I = not(\|\phi\|_w^I)$
5. $\|\phi \Rightarrow \psi\|_w^I =$ $all(\{v_1,\ldots,v_m\}, \|C_1\|_w^I \& \ldots \& \|C_n\|_w^I \rightarrow \|\psi\|_w^I)$, where $\phi = \langle\{V_1,\ldots,V_m\},\{C_1,\ldots,C_n\}\rangle$
6. $\|\langle\!\langle Id, A, V, T, Act, C, S\rangle\!\rangle\|_w^I =$ $exists(\|vars^*(Id,I)\|_w^I, do(\alpha)\ \&\ \|cstr^*(Id,I)\|_w^I)$, and $\alpha = act@id(id)@agt(a)@time(t)@tree(Id,I,w)$.
7. $\|Int[\langle V,T,C\rangle]\|_w^I = int(Holds(\|\langle V,C\rangle\|_w^I, t))$,
8. $\|\langle V,T,\mu\rangle\|_w^I = (\exists v_1 \ldots \exists v_n) holds(\|\mu\|_w^I, w, t)$, where $V = \{v_1,\ldots,v_n\}$.

The above make use of the following definitions:
$vars(Id,I) = V, \langle\!\langle Id, \_, V, \_, \_, \_, \_\rangle\!\rangle \in I$
$cstr(Id,I) = C, \langle\!\langle Id, \_, \_, \_, \_, C, \_\rangle\!\rangle \in I$
$vars^*(Id,I) = \bigcup_{s \in subs^*(Id,I)} vars(s,I)$
$cstr^*(Id,I) = \bigcup_{s \in subs^*(Id,I)} cstr(Id,I)$

Figure 2: Translation from DIS to FOL.

set of meta-language functions that stand for predicates and functions in the object language. In addition, $\mathcal{L}$ includes a single predicate symbol, *holds*, that ranges over terms, worlds and times: $holds(p,w,t)$. The term p can also have the complex form $int(Holds(q,t'))$—with upper-case term *Holds*. We use abstract syntax for logical operators in $\mathcal{L}$; thus, $holds(p\ \&\ q,w,t) \equiv (holds(p,w,t) \wedge holds(q,w,t))$, $holds(p \leftrightarrow q,w,t) \equiv (holds(p,w,t) \equiv holds(q,w,t))$, and $holds(not(p),w,t) \equiv \neg holds(p,w,t)$. In addition, if V is a set, $\{v_1,\ldots,v_n\}$, we write $exists(V,\phi)$ as shorthand for $exists(v_1,\ldots,exists(v_n,\phi)\ldots)$. To report that act-type α is performed by doing act-type β, we write: $do(\alpha@method(\beta))$.

Figure 2 gives the semantics of DISs. It assumes that all variables declared in (sub-)actions are unique as well as cross-world identity for constants, terms, predicate and function names. Possible worlds reflect alternative futures for intentions. We assume that there is a function, D, that takes a name in the object language and returns the corresponding name in the meta-language. These assumptions lead to the constraints $D(T) = t, D(P) = p$, etc.. Object-language elements will be in upper case and meta-language elements in lower case. We add the axioms:

$$holds(do(exists(v,do(e))),w,t) \equiv holds(exists(v,do(e)),w,t) \tag{1}$$

$$holds(do(not(x)),w,t) \equiv \neg holds(do(x),w,t) \tag{2}$$

$$holds(do(\alpha@time(t)),w,t') \equiv holds(do(\alpha),w,t) \tag{3}$$

We extend $\|.\|$ to any intention base, IS:

$$\|IS\|_w = \{\|I\|_w^I \mid I \in IS\}$$

We require that intentions be consistent: for any intention base, IS, it is not the case that both ϕ and $\neg\phi \in \|IS\|_{w,t}^I$. The semantics for intention is in FOL; we reify possible worlds, adopting modal logic System K (Chellas, 1980) where $acc_i(.,.,.,.)$ is a serial accessibility relation:

$$holds(int(a,Holds(p,t')),w,t) \equiv \forall w'.acc_i(a,w,w',t) \supset holds(p,w',t')$$

Here is an example of the FOL translation after utterance (5) (the θ_i's correspond to the act types in the Idi - see Appendix):

$$\begin{aligned}&holds(int(Holds(exists(\{t,x,t_t,t_r\}\\&\quad do(\theta_1@id(id_1)@method(\theta_2@id(id_2))\\&\quad @method(\theta_4@id(id_4))@method(\theta_5@id(id_5))\\&\qquad @method(\theta_3@id(id_3)))\\&\quad \&\ restaurant(x)\ \&\ meeting(m)\\&\quad \&\ near(x,h)\ \&\ italian(x)\ \&\ gt(t,t_3)\\&\quad \&\ gt(t_r,t)\ \&\ gt(t_t,t_r)),t_p)),w_0,t_3).\end{aligned} \tag{4}$$

4 Intention revision

Most approaches to belief revision are founded on the idea of *minimal change*: to revise a set of beliefs, S, with some new p, where p is inconsistent with S, one should make the minimal change necessary to S to accommodate p. Our approach is *syntactic*, assigning greater significance to formulas, and their syntactic form, that appear in a *belief or intention* base (Nebel, 1989; Ortiz, 1999) than to the consequential closure of the corresponding base (the resulting *belief set*). Intention revision takes place in two steps within this framework as follows. Let S be an agent's current set of intentions. We translate S into its *predicate form* that explicitly refers to components of a DIS so that they can be modified according to the minimality criteria above. We use the same meta level language for constants and terms as in $\mathcal{L}$ above, augmented with special predicates to name the components of a DIS. If $\langle V_c, T_c, int[\langle V_t, T_t, \langle\!\langle Id_r, A_r, V_r, T_r, E_r, C_r, S_r\rangle\!\rangle\rangle]\rangle$ is a DIS, then its translation, for all

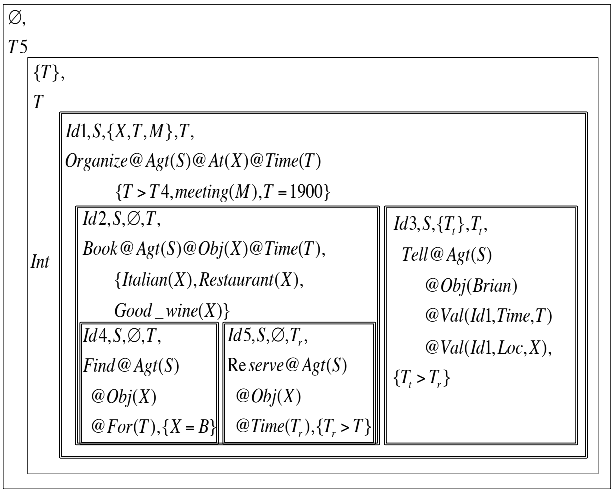


Figure 3: "And I'd like one with a good wine list."

$K_r \in C_r$, $U_r \in V_r, U_c \in V_c$, $B_r \in S_r$ is $\{id_r(id_r), agent_r(id_r, a_r), var_r(id_r, u_r), time_r(id_r, t_r), act_r(id_r, e_r), constr_r(id_r, k_r), sub_r(id_r, b_r), var_t(id_r, v_t), time_t(id_r, t_t), var_c(id_r, u_c), time_c(id_r, t_c)\} \cup \underline{nodes(ISB)}$; the latter is the set of predicate forms for each S_r.

We call a collection of DISs plus associated plan nodes an *intention base* (IB). Given an intention base, ISB, we write $\underline{ISB}$ for the translation to predicate form and $\overline{ISB'}$ for the translation into canonical form of an intention base ISB' in predicate form.[3] Let S stand for an IB; to revise S with some ϕ we create a set of equivalence classes on S: $\{S_1, S_2, \ldots, S_n\}$ such that S_1 corresponds to those elements of S that are most important and S_n to those that are least important. To revise an intention base with some ϕ, we start with ϕ and add as much of each S_i that is consistent. Revisions involve either the addition or removal of (sub)actions or constraints from or to an IB.

The appendix provides the formal definition for intention revision and a derivation of the transformation between intention structures corresponding to some of the possible continuations of our target dialog. Here, we present the general idea using the box notation for canonical DISs. The purpose of partitioning the DIS boxes is to inform the revision process. A new utterance, u, is to be interpreted as is done in plan-based theories, as contributing somehow to the current intentional structure of the discourse, which in turn is closely related to the task structure or, in our case, to the DIS. To situate (the logical form of) u correctly in the current DIS (and, hence, the current discourse structure), we consider the contents of each box followed by each sub-box. If u corresponds to a new constraint, then we check for consistency first in the outer-most box. If it is consistent then we proceed to the sub-boxes. If it is not, we revise that sub-box with the new constraint. Similarly, if u corresponds to a new action and is consistent with the current box then we continue; if not, we delete that box and all of the sub-boxes which depended on it. These guidelines can be formalized to model the intention revision process using methods from the belief revision literature, as long as we operate on the DIS predicate form.

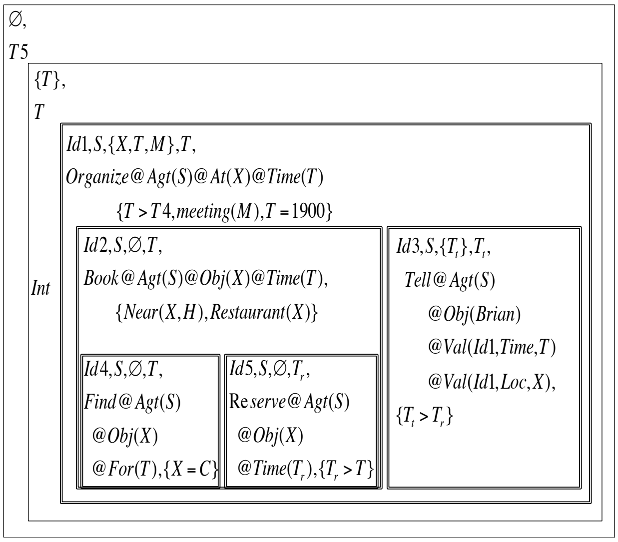


Figure 4: "Reserve a table at Chevy's instead."

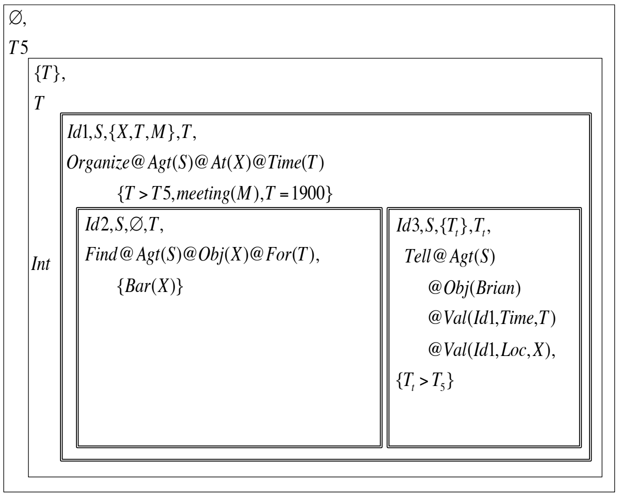


Figure 5: ". . . let's just go to a place for drinks."

Figure 3 illustrates the transformation that takes place after utterance (8a). The formula

---

[3] See (Ortiz and Hunsberger, 2013) for details.

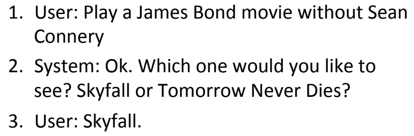
$good\_wine(X)$ is consistent with the contents of $Id2$. However, box $Id4$ and $Id2$ are not jointly consistent with the constraint $X = Z$ (shown in Figure 1). The maximal subset that is consistent with the new constraint is one that contains all of the existing constraints except $X = Z$ and $near(x, h)$. That is therefore deleted (assuming $near(x, h)$ has lowest priority) and replaced with the new constraint. Consistency is checked on the FOL translation (4).

Continuation (8b), results in the DIS shown in Figure 4. The new constraint is not consistent with the constraint in $Id4$ related to the wine; hence, the latter is deleted. Next, the new constraint is not consistent with the choice $X = B$ in Figure 1; hence, it is replaced by the new one, $X = C$. As a final example, (8d), is inconsistent, by appeal to pragmatic world knowledge - one doesn't go to restaurants just for drinks – with all of $Id2$. In the formal definition of $\underline{ISB}$, if a negated action is in ISB then the box (and sub-boxes) corresponding to it is deleted. Hence, $Id2$ and all of its sub-boxes are deleted and the new, $bar(X)$ constraint is added (Figure 5).[4] Note that, in all of these possible continuations, the side-effects to the $Id3$ component does not have to be modified. The desired changes result simply because of the way that the intention is structured and the locality of variables.

5 Implementation

We are developing a collaborative dialogue manager (CDM) that embodies the ideas described in this paper. We are testing it in a living room setting where the user asks a TV equipped with speech recognition software and natural language (NL) understanding for help in, for example, locating, playing or recording entertainment available from different content providers. CDM is an extension of Disco (Rich and Sidner, 2012), an open source dialogue development framework based on Collaborative Discourse Theory (Grosz and Sidner, 1986; Grosz and Kraus, 1996; Lochbaum, 1998). It views a VPA dialogue as a process of plan augmentation, where the purpose of the dialogue is for the system and the user to collaborate on a complete SharedPlan to meet a user's inferred intention. Each user utterance is processed by an NL pipeline consisting of named entity recognition (NER) followed by morphological, syntactic and semantic processing. The CDM then initiates a planning process by first accessing a recipe library consisting of, essentially, a collection of hierarcical networks (HTNs) that decompose high-level task (goal) structures; the recipes are written in the the ANSI/CEA-2018 standard (Rich, 2009). If a plan cannot be constructed, then one of several builtin utterance generation rules is fired and a system utterance is generated in order to acquire the necessary information from a user to further the planning process. The cycle continues until a complete plan is formed for the user's intention.

We have extended CDM with the DIS framework. CDM provides the procedural, stack-based management of *attentional state* (the "in-focus" portion of the DIS) and the dialogue segmentation. CDM generates either group-level (an intention that a group — e.g., the system and the user — perform some group action) or individual-level DISs (Hunsberger and Ortiz, 2008). The DISs depicted below include fields for two types of variables: *ExVars* and *DefVars*, those that the intending agent is free to assign values to and those determined by some other agent, respectively. There are three points where a DIS may be generated or updated: when a user utterance is interpreted, during the plan generation and decomposition process, or when a system utterance is generated. The plan augmentation process makes use of DISs directly. We will use the following simple dialogue (Figure 6) to illustrate the operation of the CDM. The "boxes" in the figures are added for readability only; the system only currently produces ASCII text with the explicit references to sub-boxes shown in the figures.



1. User: Play a James Bond movie without Sean Connery
2. System: Ok. Which one would you like to see? Skyfall or Tomorrow Never Dies?
3. User: Skyfall.

Figure 6: The James Bond example

At the start of the dialogue, CDM generates a group intention for the system to display a movie, m, for the user to watch that meets the constraints, i.e., a James Bond movie without Sean Connery. (The Group DIS, idG, is shown in Figure 7).

The system then identifies a relevant recipe from its recipe library. The top level recipe *Com-*

---

[4] Note: in the actual implementation the user is *asked* before deleting the $near(x, h)$ constraint because the meeting node has a location property as well.

```
ID/Agt/Grp: idG/g/GROUP
ExVars: m, s, j
ActType: Display@agt(System)@obj(m)
Conds: VideoConceptualWork(m),
CharacterinConceptualWork(j, m),
~VideoConceptualWorkActors(m, s), NarrativeRole(j),
Name(s, "Sean Connery"), Name(j, "James Bond"),
Person(s), Male(s)
```

Figure 7: DIS generated after user utterance 1

mandPlayVCW (Figure 8) is chosen to fulfill the group intention; the system adopts it as its own intention. Consequently, a new DIS for the system idS is generated and the group DIS idG is updated to have idS as its subBox (Figure 9). During plan decomposition, CDM generates two subBoxes $idSLookup$ and $idSPlay$ for the system DIS idS to reflect the decomposed steps (*Lookup* and *PlayVCW*) in the *CommandPlayVCW* recipe (Figure 10).

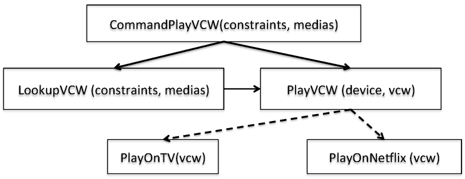


Figure 8: Recipe CommandPlayVCW

```
ID/Agt/Grp: idG/g/GROUP
ExVars: s, j
DefVars: (m, {Do(CommandPlayVCW@agt(System)@obj(_), …},
System, idS, ms)
ActType: Display@agt(System)@obj(m)
Conds: …
SubBoxes: idS

ID/Agt: idS/System
ExVars: ms
ActType: CommandPlayVCW@obj(ms)
Conds: VideoConceptualWork(ms),
CharacterinConceptualWork(j, ms),
~VideoConceptualWorkActors(ms, s)
```

Figure 9: DIS's updated user utterance 1

The *Lookup* method returns two movies *Casino Royale* and *A View to a Kill*, and the *Ask.Which* utterance generation rule is fired, producing the system utterance 2 of Figure 6. The CDM updates the system's DISs and generates a new DIS idU for the user to reflect that the movie, ms, to be played depends on the user's selection (Figure 11).

When the user chooses *Skyfall*, the system interprets that as a response to the *Ask.Which* question, and integrates the new user contribution into the user DIS *idU* by updating the ExVar mu to DefVar (Figure 12). This plan refining process continues until a complete plan is formed.

```
ID/Agt: idS/System
ExVars: ms
DefVars: (msSet, {Do(Lookup@results(_), …}, System,
idSLookup, msLookupSet)
ActType: CommandPlayVCW@obj(ms)
Conds: FORALL ms IN msSet, …
SubBoxes: idSLookup, idSPlay

ID/Agt: idSLookup/System
ExVars: msLookupSet
ActType: Lookup@results(msLookupSet)
Conds: FORALL msLookup IN msLookupSet,
VideoConceptualWork(msLookup),
CharacterinConceptualWork(j, msLookup),
~VideoConceptualWorkActors(msLookup, s)

ID/Agt: idSPlay/System
ActType:PlayVCW@obj(ms)
```

Figure 10: DIS's updated after plan decomposition

```
ID/Agt: idS/System
DefVars: (ms, {Sel(User,_,"ms", idS),_ IN msSet},User, idU,mu),
         (msSet, {Do(Lookup@results(_), …}, System, idSLookup,
            msLookupSet)
ActType: CommandPlayVCW@obj(ms)
SubBoxes: idSLookup, idSPlay, idU

ID/Agt: idSLookup/System
ExVars: msLookupSet
ActType: Lookup@results(msLookupSet)
Conds: FORALL msLookup IN msLookupSet,
VideoConceptualWork(msLookup),
CharacterinConceptualWork(j, msLookup),
~VideoConceptualWorkActors(msLookup, s)

ID/Agt: idSPlay/System
ActType:PlayVCW@obj(ms)

ID/Agt: idU/User
ExVars: mu
Conds: Sel(User, mu, "ms", idS),
            mu IN msSet
```

Figure 11: DIS's updated after CDM generates system utterance 2

In CDM, the utterance interpretation and generation processes as well as the plan decomposition and refinement processes are grounded in the DIS framework. If we continue the dialogue of Figure 6 with the user utterance "How about one with Ursula Andress," the current CDM requires a separate external consistency check; we are currently integrating that process into the system.

6 Summary and related work

We have focussed on re-planning dialogues common to scenarios involving a user and a personal assistant. In such settings (in contrast to, say, master-apprentice dialogues) a user often does not know whether a set of proposed constraints on a task will be satisfiable: new suggested constraints can conflict with already expressed ones. Further, the process of discourse segmentation can interact in a negative way with the constraint relaxation process, resulting in incorrectly situating a new utterance within a segmented discourse.

```
ID/Agt: idS/System
DefVars: (ms, {Sel(User,_,“ms”,idS), _ IN msSet},User,idU,mu),
         (msSet, {Do(Lookup@results(_), …}, System, idSLookup,
             msLookupSet)
ActType: CommandPlayVCW@obj(ms)
SubBoxes: IdSLookup

 ID/Agt: idSLookup/System
 ExVars: msLookupSet
 ActType: Lookup@results(msLookupSet)
 Conds:  FORALL msLookup IN msLookupSet,
 VideoConceptualWork(msLookup),
 CharacterinConceptualWork(j, msLookup),
 ~VideoConceptualWorkActors(msLookup, s)

 ID/Agt: idSPlay/System
 ActType:PlayVCW@obj(ms)

 ID/Agt: idU/User
 DefVars: (mu, Skyfall)
 Conds: Sel(User,mu,“ms”,idS),
             mu IN msSet
```

Figure 12: DIS's updated after user utterance 2

Such dialogues require some non-monotonic form of intention revision during the process of accommodating a new utterance into the existing dialog. We applied the DIS framework developed to model intention revision in rational agents. Intentions are structured to inform an incremental revision process: rather than completely eliminating any conflicting intention, the approach first attempts to minimally revise the contents of an individual intention; in the process, side-effects are automatically handled. Since DISs are based on a dynamic logic approach similar to DRT, a bridge is created between plan-based dialogue approaches and rigorous accounts to discourse meaning found in DRT and argued to be missing from cognitive approaches (Asher and Lascarides, 2003).

Segmented Discourse Representation Structures (SDRS) structure discourses using discourse relations; however, the rich and revisable hierarchical intention structures that we have argued for are absent. Neither DRT nor SDRS deal with the revision of structures in the case of inconsistencies. Recent work has examined the modification of decision theoretic agent preferences during dialogue (Cadilhac et al., 2011). However, their plan-correction methods do not deal with side-effects nor are they tightly linked to a formal representation of intentions. In addition, desires in the theory of SharedPlans, on which we are basing our work, formalizes desires instead as potential-intentions-to perform some action. The Collagen system maintained a segmented history of the dialogue which a user could manually examine and manipulate (Rich and Sidner, 1998): a user could retract, say, an action in a recipe plan tree and a truth maintenance system would then retract logical dependencies. Our system instead performs such "undos" automatically.

Work on correction and denials that retracts contextual information appearing earlier in a discourse is related (van Leusen, 2004; Maier and van der Sandt, 2003). That work differs, however, in that corrections and denials are explicit and discourses are not structured into larger segments. Work in SDRS in this area has not dealt with the problem of revision (Lascarides and Asher, 2009). Finally, user-initiated correction dialogs (Lochbaum, 1998)) are somewhat different as they are triggered by an observed plan obstacle.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

A. Cadilhac, N. Asher, F. Benamara, and A. Lascarides. 2011. Commitments to preferences in dialogue. In *Meeting of the SIG on Discourse and Dialogue*.

Brian F. Chellas. 1980. *Modal Logic: An Introduction*. Cambridge University Press.

Barbara J. Grosz and Sarit Kraus. 1996. Collaborative plans for complex group action. *Artificial Intelligence*, 86(1):269–357.

Barbara J. Grosz and Candace Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Luke Hunsberger and Charles Ortiz. 2008. Dynamic Intention Structures I: A theory of intention representation. *Autonomous Agents and Multiagent Systems*, pages 298–326.

Alex Lascarides and Nicholas Asher. 2009. Agreement, disputes and commitments. *Journal of Semantics*, 26(2):109–158.

Karen E. Lochbaum. 1998. A collaborative planning model of intentional structure. *Computational Linguistics*, 34(4):525–572.

E. Maier and R. van der Sandt. 2003. Denial and correction in layered DRT. In *Proceedings of diaBruck*.

Robert C. Moore. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*. Ablex Publishing Corporation.

Bernhard Nebel. 1989. A knowledge level analysis of belief revision. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 301–311.

Charles L. Ortiz and Luke Hunsberger. 2013. On the revision of dynamic intention structures. In *Eleventh International Symposium on Logical Formalizations of Commonsense Reasoning*.

Charles L. Ortiz. 1999. Explanatory update theory: Applications of counterfactual reasoning to causation. *Artificial Intelligence*, 108:125–178.

Charles Rich and Candace L. Sidner. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3/4):315–350.

Charles Rich and Candace Sidner. 2012. Using collaborative discourse theory to partially automated dialogue tree authoring. In *14th International Conference on Intelligent Virtual Agents*.

Charles Rich. 2009. Building task-based user interfaces with ANSI/CEA-2018. *IEEE Computer*, 43(8):20–27.

Noor van Leusen. 2004. Incompatibility in context: a diagnosis of correction. *Journal of Semantics*, 21(4):415–415.

Appendix: Worked out formal example

Definition 1 (Intention revision) *Let I and I' be DISs in predicate form and let S_i be the set of induced equivalence classes on I, $i \geq 1$. The prioritized removal of elements of I that conflict with $\neg\|I'\|$, which we write as $I \bullet I'$, is (Nebel, 1989):*

$$\begin{aligned} I \bullet I' = \{Y \subseteq I \mid \|\overline{Y}\| \not\vdash \neg\|\overline{I'}\|,\\ Y = \cup_i Y_i, i \geq 1\\ \forall i \geq 1 : (Y_i \subseteq S_i,\\ \forall X : Y_i \subset X \subseteq S_i \rightarrow\\ (\bigcup_{j=1}^{i-1} Y_j \cup X) \vdash \neg\|\overline{I'}\|)\} \end{aligned}$$

We can define the operation of intention revision by some I' that is inconsistent with I as:

$$I \star I' = \cap_{(Y \in I \bullet I')} \cup I'$$

Starting with I' is first augmented with the maximal subset of S_1 that is consistent (via the translation to FOL). This is repeated for each maximal subset of the next equivalence class until no additional elements of S can be consistently added.

We consider five steps, at times $t_1 < t_2 < \ldots < t_5$, of intention formation. For any t_i, the canonical form of the IB is $IS(t_i)$ and $IS[t'/t]$ indicates that all instances of t in IS are substituted by t'.

Step 1. The system (s) intends at time t_1 to organize a meeting later at $t = 1900$ (7pm). (We collapse utterances (1-3) in this step.)

$$\begin{aligned} IS(t_1) = \{\langle\emptyset, T_1, Int[\langle\{T\}, T,\\ \langle\!\langle Id_1, S, \{X, M, T\}, T, \Theta_1, \{T > T_1,\\ Meeting(M), T = 1900\}, \emptyset\rangle\!\rangle\rangle]\rangle\} \end{aligned}$$

s.t., $\Theta_1 = Organize@Agt(S)@At(X)@Time(T)$.

The predicate form of this intention is:

$$\begin{aligned} \underline{IS(t_1)} = \{\ & id_r(id_1), agt_r(id_1, s), var_r(id_1, m),\\ & time_c(id_1, t_1), time_t(id_1, t), time_r(id_1, t),\\ & var(id_1, x), act_r(id_1, \theta_1), var_t(id_1, t),\\ & constr_r(id_1, gt(t, t_1)), constr_r(id_1, eq(t, 1900))\\ & constr_r(id_1, meeting(m)\ \} \end{aligned}$$

s.t., $\theta_1 = organize@agt(s)@at(x)@time(t)$ and $gt(t, t_1)$ is the metalanguage form of $T > T_1$.

The FOL form, relative to the real world, w_0, is:

$$\begin{aligned} \|IS(t_1)\|_{w_0} = holds(int(Holds(exists(\{t, x, m\},\\ do(\theta_1@id(id_1))\ \&\ gt(t, t_1)\ \&\ eq(t, 1900)\\ \&\ meeting(m)), t)), w_0, t_1) \end{aligned}$$

Step 2. The meeting is organized by booking a restaurant near the hotel (H) and telling Brian. The "tell" action sends parameter values to Brian:

$$\begin{aligned} \underline{IS(t_2)} = \underline{IS(t_1)}[t_2/t_1] \star \{ & id(id_2), id(id_3),\\ & sub_r(id_1, id_2), agt(id_2, s), time(id_2, t),\\ & sub_r(id_1, id_3), agt(id_3, s), time(id_3, t_t),\\ & var(id_3, t_t), act(id_2, \theta_2), act(id_3, \theta_3),\\ & constr(id_2, restaurant(x)), constr(id_2, near(x, h)),\\ & constr(id_2, italian(x), constr(id_3, gt(t_t, t))\} \end{aligned}$$

where $\theta_2 = book@agt(s)@obj(x)@time(t)$ and $\theta_3 = tell@agt(s)@obj(brian)@val(id_1, time, t)$ $@val(id_1, loc, x)$

In canonical form we have:

$$\begin{aligned} IS(t_2) = \{ & \langle\emptyset, T_2, Int[\langle\{T\}, T, \langle\!\langle Id_1, S, \{X, M, T\}, T,\\ & \Theta_1, \{T > T_2, Meeting(M), T = 1900\}, \{Id_2, Id_3\}\rangle\!\rangle\rangle]\rangle,\\ & \langle\!\langle Id_2, S, \emptyset, T, \Theta_2, \{Restaurant(X),\\ & \quad Near(X, H), Italian(X)\}, \emptyset\rangle\!\rangle,\\ & \langle\!\langle Id_3, S, \{T_t\}, T_t, \Theta_3, \{T_t > T\}, \emptyset\rangle\!\rangle\} \end{aligned}$$

where $\Theta_2 = Book@Agt(S)@Obj(X)@Time(T)$ and: $\Theta_3 = Tell@Agt(S)@Obj(Brian)$ $@Val(Id_1, Time, T)@Val(Id_1, Loc, X)$.

Step 3. The system decides to book the table by finding and reserving a restaurant.

$$\begin{aligned} \underline{IS(t_3)} = \underline{IS(t_2)}[t_3/t_2] \star \{ & id(id_4), id(id_5),\\ & sub(id_2, id_4), sub(id_2, id_5), agt(id_4, s),\\ & agt(id_5, s), time(id_4, t), time(id_5, t_r),\\ & act(id_4, \theta_4), act(id_5, \theta_5), constr(id_5, gt(t_r, t))\} \end{aligned}$$

where $\theta_4 = find@agt(s)@obj(x)@for(t)$ and $\theta_5 = reserve@agt(s)@obj(x)@time(t_r)$.

In canonical form, the result is:

$$\begin{aligned}
&IS(t_3) = \{\langle\emptyset, T_3, Int[\langle\{T\}, T,\\
&\quad \langle\!\langle Id_1, S, \{X, M, T\}, T, \Theta_1, \{T > T3,\\
&\quad Meeting(M), T = 1900\}, \{Id_2, Id_3\}\rangle\!\rangle\rangle]\rangle,\\
&\quad \langle\!\langle Id_2, S, \emptyset, T, \Theta_2, \{Restaurant(X),\\
&\quad Near(X, H), Italian(X)\}, \{Id_4, Id_5\}\rangle\!\rangle,\\
&\quad \langle\!\langle Id_3, S, \{T_t\}, T_t, \Theta_3, \{T_t > T, T > T_2\}, \emptyset\rangle\!\rangle,\\
&\quad \langle\!\langle Id_4, S, \emptyset, T, \Theta_4, \emptyset, \emptyset\rangle\!\rangle,\\
&\quad \langle\!\langle Id_5, S, \emptyset, T_r, \Theta_5, \{T_r > T\}, \emptyset\}\\
&\text{s.t., } \Theta_4 = Find@Agt(S)@Obj(X)@For(T),\\
&\Theta_5 = Reserve@Agt(S)@Obj(X)@Time(T_r)
\end{aligned}$$

The FOL translation is given by formula (4).

Step 4. The user selects Zingari (Z) and the system adds it to the intention structure.

$$\underline{IS(t_4)} = \underline{IS(t_3)}[t_4/t_3] \star constr(id_4, eq(x, z))$$

The canonical form is given in Figure 1. It follows that the system also intends to tell Brian of the location, Zingari, and to reserve a table there.

Step 5. The system revises its intention to include the constraint of a good wine list. (Previously, "$\star$" corresponded to set union) We revise the intention and assume a joint user-system selection of Barbacco ($x = b$). The knowledge base also contains, with highest priority, the following:

$$\begin{gathered}
holds(good\_wine(y) \leftrightarrow eq(y, b), w, t) \qquad (5)\\
holds(near(z, h) \;\&\; \sim near(b, h), w, t)\\
holds(italian(z) \;\&\; italian(b), w, t)\\
holds(restaurant(z) \;\&\; restaurant(b)w, t)
\end{gathered}$$

We have,

$$\begin{aligned}
&\underline{IS(t_5)} =\\
&\quad IS(t_4)[t_5/t_4] \star \{constr(id_4, good\_wine(x))\}
\end{aligned}$$

The following are the set of priority classes that we will use, separating the tree, constraints and assignments. General or more detailed rules can be written (Ortiz and Hunsberger, 2013).

$$\begin{aligned}
&S_1(IS(t_4)) = \{id_r(id_1), agt_r(id_1, s), var_r(id_1, m),\\
&\quad var_r(id_1, x), var_t(id_1, t), act_r(id_1, \theta_1),\\
&\quad time_c(id_1, t_4), time_t(id_1, t), time_r(id_1, t),\\
&\quad id(id_2), sub_r(id_1, id_2), agt(id_2, s), sub(id_2, id_5),\\
&\quad id(id_3), sub_r(id_1, id_3), agt(id_3, s), id(id_5),\\
&\quad id(id_4), sub(id_2, id_4), agt(id_4, s), agt(id_5, s),\\
&\quad time(id_2, t), time(id_3, t_t), time(id_4, t), time(id_5, t_r),\\
&\quad act(id_2, \theta_2), act(id_3, \theta_3), act(id_4, \theta_4),\\
&\quad var(id_3, t_t), act(id_5, \theta_5)\}\\
&S_2(IS(t_4)) = \{constr(id_3, gt(t_t, t_r)),\\
&\quad constr(id_5, gt(t_r, t))\\
&\quad constr(id_2, restaurant(x)), constr(id_2, italian(x)),\\
&\quad constr_r(id_1, gt(t, t_4)), constr_r(id_1, meeting(x)),\\
&\quad constr(id_3, gt(t_t, t)), constr(id_5, gt(t_r, t))\},\\
&S_3(IS(t_4)) = \{constr(id_4, eq(x, z)),\\
&\quad constr(id_1, eq(t = 1900))\}\\
&S_4(IS(t_4)) = \{constr(id_2, near(x, h))\}
\end{aligned}$$

S_1 and S_2 go through but $constr(id_4, eq(x, z))$ (S_3) and S_4 conflict and are not included in $\underline{IS(t_5)}$. To see this, we translate to FOL, and apply axiom (1):

$$\begin{aligned}
&\|IS(t_4)\|_{w_0} =\\
&\quad holds(int(Holds(exists(\{t, x, t_t, t_r, m\}\\
&\quad do(\theta_1@id(id_1)@method(\theta_2@id(id_2)\\
&\quad @method(\theta_4@id(id_4))@method(\theta_5@id(id_5))\\
&\qquad @method(\theta_3@id(id_3)))\\
&\quad \& \; restaurant(x) \;\&\; gt(t_t, t) \;\&\; gt(t, t_3)\\
&\quad \& \; italian(x) \;\&\; near(x, h) \;\&\; eq(t, 1900)\\
&\quad \& \; gt(t_r, t) \;\&\; gt(t_t, t_r) \;\&\; good\_wine(x)\\
&\quad \& \; meeting(m) \;\&\; eq(x, z))))), w_0, t_4).
\end{aligned}$$

We eliminate *holds* expressions by referring to the accessibility relation and (1), converting "&" to conjunction. The result is inconsistent, given axioms (5). Similarly, S_4, is also inconsistent. Barbacco can now be inserted into the DIS. The result (shown in Figure 3). It follows that the system will tell Brian that the location is Barbacco, as desired.

The remaining cases are handled similarly. In choosing a bar, an axiom would preclude that together with booking a restaurant; by S_1 and the mapping back to canonical form, we would have an inconsistency, retracting the entire "box" for $Id2$.