Domain-Independent Discourse Ontology for Utterance Interpretation and Generation in Human-Computer Dialogue

Vladimir Popescu[1, 2] and **Jean Caelen**[1] and **Corneliu Burileanu**[2]

Abstract. This paper is concerned with the design, development and usage of a domain- and language-independent discourse ontology, useful for utterance interpretation and generation in service-oriented human-computer dialogue. Although the syntagm "discourse ontology" is not new, current work assimilates this term to rather domain-specific ontologies embedding discourse-related knowledge. In contrast with these approaches, a generic task-independent discourse ontology is proposed here; this knowledge base, structured in an object-oriented paradigm formalized in a first-order logic, can be used for rhetorical structure computation in human-computer dialogue. In this paper we will show how this ontology is structured, emphasizing the duality between physical structure (in the form of an XML database) and logical structure (enclosed in a set of axioms). Then, it will be shown how this ontology can be used for rhetorical structure updating in the process of (machine) utterance generation for human-computer dialogue.

1 INTRODUCTION

The research described in this paper concerns the design and development of a generic (i.e., task-independent) architecture for service-oriented spoken dialogue systems [2]. More specifically, we are concerned with the benefits that rhetorical structuring can bring to both utterance interpretation [2] and generation [6]. In order to separate task-specific issues (for example, for book reservation in a library [7]) from domain-independent but discourse-related aspects, we distinguish task-related knowledge (that are structured in a task ontology) from general, task-independent but discourse-specific elements (such as rhetorical relations between spans of text).

Concerning the task ontology, this can be automatically built in a data-driven approach, using a Wizard of Oz-acquired dialogue corpus to drive appropriate selection of concepts and relations from a general ontology such as Word Net [9]. The interest in building this ontology automatically is motivated by the need to re-built task-specific knowledge bases for each new application domain.

As for the discourse ontology, given its genericity (i.e., domain independence), it can be used for any task in any application domain, this is why building it by hand is a tractable process.

Thus, in this paper we present the details of the discourse ontology that was built for utterance interpretation and generation in dialogue. In fact, this research represents a step further towards generic dialogue modelling, continuing previous research in this respect: from a task-independent dialogue game ontology specifying a hierarchy of dialogue types and sessions (viz. opening, information exchange, closing) [2] used at the dialogue planning level, we moved to a dialogue and domain-independent **discourse** ontology, which is in the focus of this paper.

The dialogue-independent discourse ontology is built with particular emphasis on *temporal* aspects in dialogue, having a dual structure:

- **physical** consistence – rendered concrete in an XML database;
- **logical** organization – quantified via several *axioms*.

In this paper we will put forward the logical organization of the discourse ontology, however showing, when needed, dependencies between the two aspects of its structure. Thus, the axioms stating the logical structure of the discourse ontology are of several types: (i) *meronymic* (hierarchical) axioms, stating inheritance relations between entities, (ii) semantic (compositional) axioms, formalizing either temporal aspects in dialogue, or rhetorical relations semantics, or auxiliary features (refer to Section 2 for details). An important point to put forward is that the discourse ontology is mainly an *actions* knowledge base, that is, the entities in the ontologies are mainly predicates or functions formalizing operations, taking **utterances** as atomic arguments.

Concerning related work in designing or implementing discourse ontologies, we can emphasize that reported in [3], concerned with a domain-specific discourse ontology in the medical domain, where only one part is domain-independent and discourse-specific (however, even in this part, several task-specific elements occur, such as predicates related to patients or Alzheimer disease). Moreover, this ontology encodes knowledge related to particular aspects of scientific discourse (e.g. a predicate for refuting an argument). Another interesting work in building discourse ontologies is reported in [4], where particular attention is paid to modelling temporal aspects in a multimodal discourse ontology for multi-party dialogue understanding; this ontology relies on the CLib library and borrows elements from the General Ontology for Linguistic Description (such as clustering of the top level entities in entities and events). Moreover, a certain form of discourse interpretation is possible, in a framework similar to Livia Polanyi's Linguistic Discourse Model (LDM), which provides only structural constraints on relations between spans on text, with no semantics associated to these relations. However, our paper differs from these approaches in that no domain-specific knowledge is embedded into the discourse ontology, which is nevertheless suited for human-computer dialogue (e.g. by providing a clustering of the

[1] LIG/GETALP, Grenoble Institute of Technology, France, email: {vladimir.popescu, jean.caelen}@imag.fr

[2] University "Politehnica" of Bucharest, Romania, email: cburileanu@messnet.pub.ro

utterances in affirmative and interrogative ones).

As for the *usage* of such a discourse ontology in spoken dialogue systems, we put forward two crucial aspects to be taken into account when aiming at *successful* dialogues [2]:

1. rhetorical structure **updating** in machine utterance generation (for details in this respect, refer to [7] and [8]);
2. dialogue **interpretation** (for details, refer to [2] for a thorough discussion on the subject).

The remainder of the paper is structured as follows: Section 2 presents in detail the discourse ontology, emphasizing at the same time methodological issues guiding design choices, and relevant implementation details; Section 3 presents an illustrative usage example for the discourse ontology, namely the updating of the rhetorical structure of the dialogue, in the process of machine utterance generation. Finally, Section 4 concludes the paper and provides a discussion on current limitations and prospects.

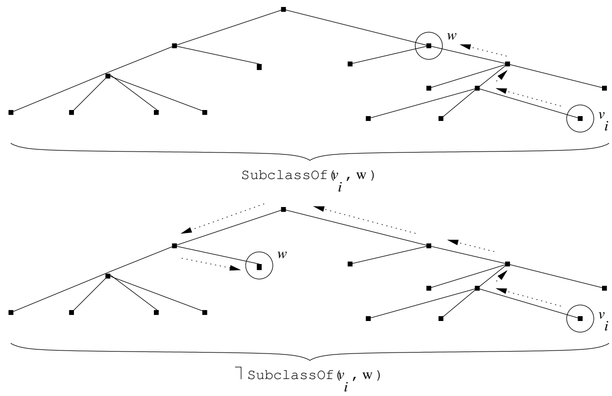

Figure 1. Meaning of the predicate SubclassOf/2

2 DISCOURSE ONTOLOGY

2.1 Principles

The discourse ontology is structured as a hierarchical knowledge base, whose items are the "ingredients" of the semantics of rhetorical relations connecting spans of text. The rhetorical relations considered are a subset of those provided in Segmented Discourse Representation Theory (SDRT) [1]. However, from SDRT only the rhetorical relations taxonomy and the *informal* semantics of these have been considered; from a formal point of view, the semantics of the rhetorical relations have been emulated in a first-order logic [7], so that a semantics-driven on-the-fly discourse structure update is possible, in the context of human-computer dialogue.

Thus, this ontology is defined via a set of axioms, specifying at first its hierarchical structure. Hence, we stipulate that each entity in the ontology is either a **table** (if it is not terminal) or a **type** (if it is terminal). Thus, knowledge is structured as a relational hierarchical database, where each type becomes table, whose types are non-null *instances* of the hierarchically upper table. The knowledge base is traversed until, for a certain given identifier *ID*, a terminal instance is found, that is, which is neither table, nor type in another table. In order to acquire, from a logical point of view, a representation which renders conspicuous this knowledge structuring, we posit several axioms, expressed in a first-order logic.

In order to specify these axioms, a set of **auxiliary** predicates are needed, for expressing order relations, either between *sets*: (i) $\in$::= MemberOf/2 , (ii) $\subset$::= SubclassOf/2 , (iii) $\ni$::= ClassOf/2 , (iv) $\supset$::= SuperclassOf/2 , (v) $\cap = \emptyset$ ::= Disjoint/2 , (vi) $\cup$ = All ::= ExhaustiveDecomposition/4 , (vii) $(\cap = \emptyset) \wedge (\cup = \text{All})$ ::= Partition/n , (viii) $\bar{\in}$ ::= SmallestMemberOf/2 , (ix) $\bar{\subset}$ ::= SmallestSubclassOf/2 [3] , or between *numbers*: (i) <::= smaller/2 , (ii) >::= greater/2 , (iii) =::= equals/2 .

These predicates are connected via the usual logic connectives: $\wedge$, $\vee$, $\neg$ and $\Rightarrow$; moreover, variables can be quantified either existentially (by $\exists$), or universally (by $\forall$). The meaning of the auxiliary predicates is stated via constraints induced relations between their arguments, in the appropriate ontology (either the discourse ontology or the task ontology). Thus, for instance the predicate SubclassOf/2 returns true for two arguments v_i and w if and only if there exists, in the ontology graph, a one-way path that does not cross more than once the same node, between these two arguments; the situation is illustrated in figure 1.

Then, inclusion relations between tables, types and instances are specified in the ontology: if $Instance_k^{(ij)}$ is of type $Type_j^{(i)}$, then MemberOf($Instance_k^{(ij)}, Type_j^{(i)}$) , when the instance is terminal, and SubclassOf($Instance_k^{(ij)}, Type_j^{(i)}$) when the instance is non-terminal.

Hence, we obtain a set of axioms **structuring** the ontology; the axioms are parameterized on the indexes of the tables, types and instances:

$\forall i,j,k : \forall l :$ (MemberOf($Instance_l^{(ijk)}, Type_k^{(ij)}$) $\wedge$ equals($Instance_l^{(ijk)}, Type_l^{(ijk)}$)) $\vee$ (SubclassOf($Instance_l^{(ijk)}, Type_k^{(ij)}$) $\wedge$ equals($Instance_l^{(ijk)}, Type_l^{(ijk)}$)) .

Then, in the same way as above, we define a set of axioms (parameterized on the indexes of the entities) stating the relations between instances and types:

- if an instance is non-terminal, then there are instances being included in the former:
  $\exists i,j,k,l :$ equals($Instance_l^{(ijk)}, Type_l^{(ijk)}$) $\Rightarrow \exists m :$ (MemberOf($Instance_m^{(ijkl)}, Type_l^{(ijk)}$)$\wedge\neg$equals($Instance_m^{(ijkl)}, Type_m^{(ijkl)}$)$\vee$ (SubclassOf($Instance_m^{(ijkl)}, Type_l^{(ijk)}$) $\wedge$ equals($Instance_m^{(ijkl)}, Type_m^{(ijkl)}$) ;
- an instance is at the same time of two types (multiple inheritance) if and only if it includes at least one instance whose identifier designates two instances, in the two types concerned:
  $\forall i,j,k :$ (MemberOf($Instance_k^{(ij)}, Type^{(ij)}$) $\wedge$ MemberOf($Instance_k^{(ij)}, Type^{(ij)\prime}$))$\vee$(SubclassOf($Instance_k^{(ij)}, Type^{(ij)}$)$\wedge$ SubclassOf($Instance_k^{(ij)}, Type^{(ij)\prime}$)) $\Leftrightarrow \exists m, m' :$ (MemberOf($Instance_m^{(ij)}, Type^{(ij)}$)$\wedge$MemberOf($Instance_{m'}^{(ij)}, Type^{(ij')\prime}$))$\vee$ (SubclassOf($Instance_m^{(ij)}, Type^{(ij)}$) $\wedge$ SubclassOf($Instance_{m'}^{(ij)}, Type^{(ij')\prime}$)) $\wedge$ equals($ID(Instance_k^{(ij)}), ID(Instance_m^{(ij)})$) $\wedge$ equals($ID(Instance_k^{(ij)}), Instance_{m'}^{(ij)}$) .

Then, a set of predicates for handling **temporal** events are stated, in terms of the auxiliary predicates defined above [10]:

1. Initiates(α, β, t) ::= $\alpha \wedge$(greater(t_α, t)$\vee$equals(t_α, t)) $\Rightarrow \beta$; this predicate states that the beginning of the event described by α at the moment t entails the validity of the clause β;
2. Terminates(α, β, t) ::= $\alpha \wedge$ (smaller(t_α, t)) $\Rightarrow \neg\beta$; this predicate states that the end of the event described by α at the moment t entails the non-validity of the clause β;

[3] The last predicate is defined via the axiom: $\forall y \forall x$: MemberOf(x, s) $\Rightarrow$ SubclassOf(x, y) $\Rightarrow$ SubclassOf(SmallestSubclassOf(s, y), y).

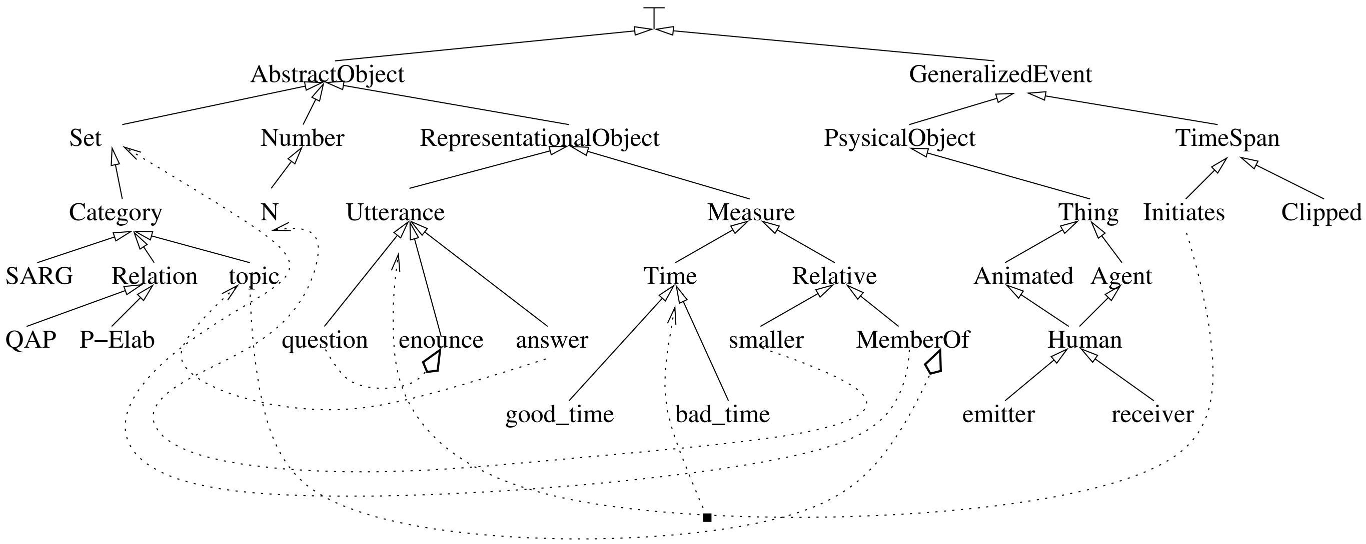

Figure 2. Discourse Ontology

3. Clipped(α, t, t') ::= $\exists\beta$: Terminates$(\beta, \alpha, t_\beta \wedge$smaller$(t, t_\beta) \wedge$smaller$(t_\beta, t')$; this predicate states that the event described by α is determined by another event at a moment τ so that $\tau \in (t, t')$.

For stating the rhetorical predicates that are relevant for expressing the semantics of the rhetorical relations used in human-computer dialogue [2], [7], we define several predicates and functions, expressed in a first-order logic. Although SDRT uses several (non-monotonic) dynamic semantics for a formal account of the discourse structure, our fundamental principle is to rely only on a first-order logic approximation or rhetorical relations meanings, thus rendering the discourse structure updating mechanism computable. Therefore, the semantics of rhetorical relations are defined in terms of terminal predicates in the discourse ontology; these predicates take utterance *labels* as arguments; these labels are then expanded into the semantics of the utterances, expressed still in a first-order logic, but using the task ontology [9].

Thus, denoting by Ω the set of entities in the task/domain ontology, by $K(\alpha)$ the logic form expressing the semantics of the utterance labeled α, by $t(\alpha)$ the moment when the utterance α is being produced, by U and M the human user and the machine, respectively, by Γ the set of speech act types [2], [8], the discourse predicates are defined as follows:

1. question(α) ::= $\exists v$: MemberOf$(v, K(\alpha)) \wedge \neg\exists\omega$: MemberOf$(\omega, \Omega) \wedge$ equals(v, ω) ;
2. enounce(α) ::= $\neg$question(α) ;
3. confirmation(β, α) ::= equals$(K(\beta), K(\alpha)) \wedge$ $\neg$equals(emitter(β), emitter(α)) ;
4. answer(β, α) ::= greater$(t(\beta), t(\alpha)) \wedge$ equals(topic(α), topic(β)) $\wedge \forall v$: MemberOf$(v, K(\beta)) \Rightarrow \exists\omega$: MemberOf$(\omega, \Omega) \wedge$ equals(v, ω) ;
5. topic(α) ::= ExhaustiveDecomposition$(i, j; v_i, \omega_j) \wedge$ MemberOf$(v_i, K(\alpha)) \wedge$ MemberOf$(\omega_j, \Omega) \wedge (\exists k$: equals$(v_k, \omega_j) \wedge$ MemberOf$(v_k, K(\alpha)))$;
6. emitter(α) ::= $(U \wedge \neg$Disjoint$(U, K(\alpha))) \vee (M \wedge \neg$Disjoint$(M, K(\alpha)))$;
7. entails(α, β) ::= $(K(\alpha) \Rightarrow K(\beta)) \wedge$ equals$(t(\alpha), t_{\not{}})$;
8. $SARG(\alpha)$ ::= $\tilde{K}(\alpha)$: SubclassOf$(\tilde{K}(\alpha), K(\alpha)) \vee$equals$(\tilde{K}(\alpha), K(\alpha)) \wedge \exists\gamma$: MemberOf$(\gamma, \Gamma) \wedge \neg$Disjoint$(\gamma, \tilde{K}(\alpha))$;
9. good_time$(\Delta t(\beta))$ ::= $\exists\alpha$: $\neg$Disjoint(topic(α), topic(β)) $\wedge$ smaller$(t(\alpha), t(\beta)) \wedge$ (SubclassOf$(\Delta t(\beta), \Delta t(\alpha)) \vee$ equals$(\Delta t(\beta), \Delta t(\alpha))$) ;
10. bad_time$(\Delta t(\beta))$ ::= $\neg$good_time$(\Delta t(\beta))$;
11. $\Delta t(\beta)$ ::= $\forall t_1, t_2$: smaller$(t_1, t_2) \wedge$ MemberOf$(t_2 \setminus t_1$, topic$(\beta)) \wedge$ MemberOf$(t_1, K(\beta)) \wedge$ MemberOf$(t_2, K(\beta))$.

In the list of axioms stated above, (i) $SARG/1$ stands for "Speech Act Related Goal" [1], a function returning the *conversational* goal contained in the utterance given as argument to this function, (ii) good_time/1 is a predicate returning true if and only if the time stated in an utterance came from a speaker is contained in the SARG contained in a previous utterance, came from a different speaker; these two predicates are rather specific for a certain type of dialogues, concerned with negotiation of time interval for resource availability (e.g. book reservation in a library [7], or meeting room reservation [2]), (iii) $\Delta t/1$ is a function returning the time interval *proposed* in the utterance taken as argument. All the predicates defined via the axioms presented above concern utterance features, as opposed to discourse features (which give an account of relations between pairs of utterances or spans of text).

As for the semantics of the 17 rhetorical relations used in human-computer dialogue, these are only specified as a set of *labels* in the discourse ontology; their semantics are specified in separate components, concerned either with dialogue *interpretation* [2], or with rhetorical structure updating in machine utterance *generation* [6], [7]. Hence, these semantics will not be presented here in detail, deferring to the next section an illustrating example concerned with utterance generation.

2.2 Synopsis

In this section we present a synthetic view of the discourse ontology, using the following notations (inspired in part by the Unified Modelling Language - UML notation for relations between objects):

- $\top$ ::= the most general concept in the ontology;

- ⇾::= the meronymic relation between two concepts: the left argument is a hyponym of the right argument;
- →::= the usage relation between two concepts: the left argument is using (i.e., taking as argument) the right argument;
- ◇− ::= the composition relation between two concepts: the right argument is taking part in the structure of the left argument;
- ⟐− ::= the *n*-ary composition relation between two concepts: the right argument, *quantified by* ∀, is taking part in the structure of the left argument;
- ◇ <::= the disjoint composition: only *one* of the (two) right arguments (separated by ∨) is taking part in the structure of the left argument.

Hence, a fragment of the discourse ontology is presented in figure 2; the four upper levels in the ontology are rather general (that is, independent of the purview of the ontology) and designed according to ideas from [10], whereas the bottom levels are discourse-related, and the entities are defined according to the principles presented in Section 1.

It can be noticed that only the bottom levels (from level four on, downwards) are discourse-related, the rest of the ontology being general, and compatible with the upper-level concept hierarchy in the dialogue games ontology [2], or in particular task ontologies [6]; thus, the discourse ontology can be used in order to guide the design of dialogue games or task ontologies. However, unlike other ontologies such as Word Net, we have not split meronymic relations in a-kind-of (AKO) and is-a (ISA) relations, so that the fact that a certain concept is an instance of a type is specified through structuring axioms; these axioms allow for a certain instance to become a type, when the granularity on the ontology needs to be extended by adding bottom (hierarchical) levels.

3 USAGE EXAMPLE: RHETORICAL STRUCTURING IN UTTERANCE GENERATION

In this section we will show how the discourse ontology presented above can be used for updating the discourse structure (represented in the framework proposed in SDRT) in the process of machine utterance generation in dialogue. This processing feature is instantiated in a rhetorical structuring component, extensively described elsewhere [7], [8]. However, the basic idea is that a subset of the rhetorical relations in SDRT are used, namely 17 relations, divided into several categories: (i) *monologic* rhetorical relations, used for computing the discourse structure of the speech turn due to be generated by the computer: Alternation, Background, Consequence, Elaboration, Contrast and Parallel, whose informal semantics are those provided by vanilla SDRT [1]; (ii) *dialogic* relations, used for appending the discourse structure of the current machine speech turn to the discourse structure of the whole dialogue: Question Elaboration (Q-Elab), Indirect Question-Answer Pair (IQAP), Plan Correction (P-Corr), Plan Elaboration (P-Elab), Background$_q$, Elaboration$_q$ (Elab$_q$), Narration$_q$, Question-Answer Pair (QAP), Acknowledgement (Ack) and Not Enough Information (NEI), whose informal semantics are provided in vanilla SDRT as well.

The discourse structure updating process consists essentially in trying to connect the current utterance (to be generated by the computer) to previous utterances in dialogue, either via monologic rhetorical relations (if the producer of the utterances is the same), or via dialogic rhetorical relations (if the utterances have different producers); thus, for each pair of utterances that includes the current utterance

M_1: Welcome to Groplan$^{\pi_{11}}$. This system allows you to find reference materials in the library of our University$^{\pi_{12}}$. At any time you can get help by saying general usage mode$^{\pi_{13}}$. What do you want$^{\pi_{14}}$?
U_1: Theatre book ancient tragedy humm... ancient Greek theatre$^{\pi_{15}}$
M_2: I have found more than 13 ancient Greek tragedies$^{\pi_{21}}$. You can give extra specifications, for instance on the author, or see the solutions or perform a new search$^{\pi_{22}}$.
U_2: Aristophanes$^{\pi_{23}}$.
M_3: There is no ancient Greek tragedy written by Aristophanes$^{\pi_{31}}$. Yet, there are written by Sophocles, Euripides or Aeschylus$^{\pi_{32}}$. Which proposal are you interested in$^{\pi_{33}}$?
U_3: A tragedy with parricide I am looking for a tragedy where the son kills his father$^{\pi_{34}}$. By Aeschylus$^{\pi_{35}}$.
M_4: There is neither any tragedy with parricide where the son murders his father, written by Aeschylus$^{\pi_{41}}$. Instead, there are, written by Sophocles or Euripides$^{\pi_{42}}$. Which proposal are you interested in$^{\pi_{43}}$?
U_4: I want to read about a son that kills his father$^{\pi_{44}}$.
M_5: I have 2 tragedies written by Sophocles$^{\pi_{51}}$. You can see the solutions or perform a new search$^{\pi_{52}}$.
U_5: See the solutions$^{\pi_{53}}$.
M_6: The first tragedy is "Oedipus the King"$^{\pi_{61}}$. It is about a prince who murders his father, becomes king and marries his mother, getting to realize and regret it in the end$^{\pi_{62}}$. Would you like to get more information, the next tragedy or to perform a new search$^{\pi_{63}}$?

Figure 3. Typical dialogue

due to be generated, each of the possible rhetorical relations is tried, and those found to hold are included in the discourse structure.

In order to illustrate the discourse structure updating process, we consider the dialogue situation where a human user (denoted by U) tries to reserve a book at a library; the librarian is represented by the computer system (denoted by M). Thus, a typical dialogue handled by the system is shown in figure 3, where subscripts in $\{U|M\}_i$ denote speech turns, π_{ij} represent utterance labels, i and j denoting the speech turn and the utterance in a speech turn, respectively.

In order to get a glimpse on the complexity of the discourse structure computed by the rhetorical structuring component for the dialogue shown in figure 3, we show it in figure 4; however, in this paper only one small part of the discourse structure updating process will be presented; for further details, please refer to [7] or [8].

Thus, we show one part of the discourse relation computation between (monologic) utterances π_{11} and π_{12}. More specifically, it will be shown how the monologic relation Alternation (abbreviated by ALT) is ruled out between these two utterances.

The process starts with reading the logic forms expressing the semantics of these two utterances; these logic forms are provided by the dialogue controller [2], [6]:

$\pi_{11} \mapsto K(\pi_{11}); \pi_{12} \mapsto K(\pi_{12})$.

Then, from the fact the the producers (emitter/1) of the two utterances coincide, we have that the set of monologic rhetorical relations must be checked:

$\mathsf{equals}(\mathsf{emitter}(\pi_{11}), \mathsf{emitter}(\pi_{12})) \Rightarrow \langle \mathsf{RhetRel} \rangle(\pi_{11}, \pi_{12}) \equiv$ monologic relation .

Hence, the Alternation discourse relation is first selected, and its semantics (denoted by Σ_{ALT}) is retrieved:

$\langle \mathsf{RhetRel} \rangle \leftarrow \mathsf{ALT} \Rightarrow \Sigma_{\langle \mathsf{RhetRel} \rangle} \leftarrow \Sigma_{\mathsf{ALT}}$.

Then, the computation follows the steps shown below:

1. The semantics of the rhetorical relation is expanded: $\Sigma_{\mathsf{ALT}} ::= \mathsf{enounce}(\pi_{11}) \wedge \mathsf{enounce}(\pi_{12}) \wedge \neg\mathsf{equals}(K(\pi_{11}) \vee K(\pi_{12}), \emptyset) \wedge \mathsf{equals}(K(\pi_{11}) \wedge K(\pi_{12}), \emptyset)$;
2. The meanings of the two utterances (denoted by $K(\pi_{ij})$) are expanded, using predicates in the task ontology [9]:

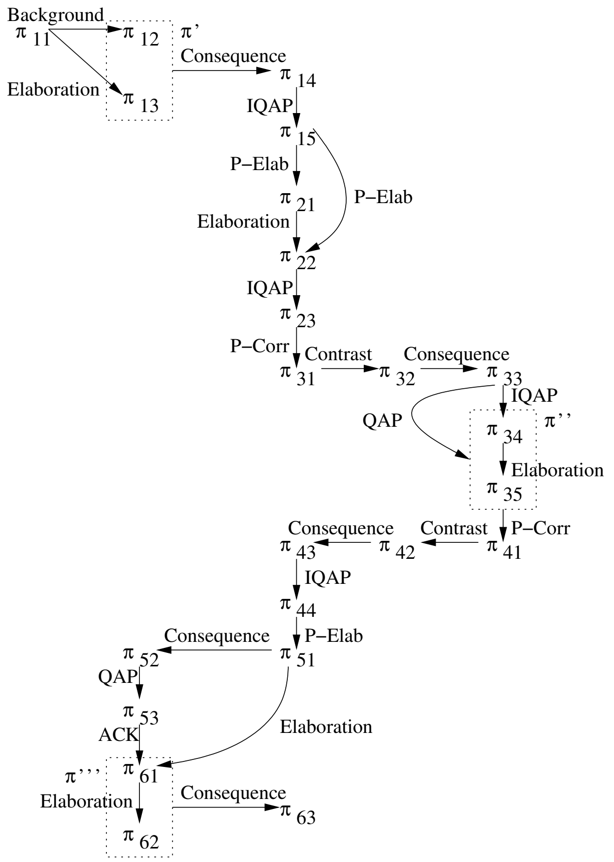

Figure 4. Discourse structure for a typical dialogue

$\Sigma_{\mathsf{ALT}}(K(\pi_{11}), K(\pi_{12})) \mapsto \mathsf{enounce}(\pi_{11}) \wedge \mathsf{enounce}(\pi_{12}) \wedge \neg\mathsf{equals}((\exists X : \mathsf{greeting}(X) \wedge \mathsf{equals}(X, \text{'groplan'})) \vee (\exists X, Y, Z, T : \mathsf{agent}(X) \wedge \mathsf{equals}(X, \text{'groplan'}) \wedge \mathsf{object}(Y) \wedge \mathsf{equals}(Y, \text{'material'}) \wedge \mathsf{find}(X, Y) \wedge \mathsf{patient}(T) \wedge \mathsf{equals}(T,$
$\neg\mathsf{emitter}(\pi_{12})) \wedge \mathsf{equals}(Z, \text{'univ\_library'}) \wedge \mathsf{MemberOf}(Y, Z) \wedge \mathsf{equals}(\Delta t(\pi_{12}), t)), \emptyset) \wedge \mathsf{equals}((\exists X : \mathsf{greeting}(X) \wedge \mathsf{equals}(X, \text{'groplan'})) \wedge (\exists X, Y, Z, T : \mathsf{agent}(X) \wedge \mathsf{equals}(X, \text{'groplan'}) \wedge \mathsf{object}(Y) \wedge \mathsf{equals}(Y, \text{'material'}) \wedge \mathsf{find}(X, Y) \wedge \mathsf{patient}(T) \wedge \mathsf{equals}(T, \neg\mathsf{emitter}(\pi_{12})) \wedge \mathsf{equals}(Z, \text{'univ\_library'}) \wedge \mathsf{MemberOf}(Y, Z) \wedge \mathsf{equals}(\Delta t(\pi_{12}), t)), \emptyset)$;

3. The truth value of each clause in the conjunctive form above is computed:
   (a) $\mathsf{enounce}(\pi_{11}) \mapsto \mathsf{TRUE}$;
   (b) $\mathsf{enounce}(\pi_{12}) \mapsto \mathsf{TRUE}$;
   (c) $K(\pi_{11}) \vee K(\pi_{12}) \mapsto \exists X, Y, Z, T : (\mathsf{greeting}(X) \vee \mathsf{agent}(X)) \wedge \mathsf{equals}(X, \text{'groplan'}) \wedge ...$:
      i. $\exists X : \mathsf{greeting}(X) \wedge \mathsf{agent}(X) \mapsto \mathsf{TRUE} \Leftrightarrow \forall X : \neg\mathsf{greeting}(X) \wedge \neg\mathsf{agent}(X) \mapsto \mathsf{FALSE}$;
      ii. $\mathsf{equals}(X, \text{'groplan'}) \Rightarrow (\mathsf{agent}(X) \wedge \mathsf{greeting}(X) \mapsto \mathsf{TRUE})$;
      iii. $\mathsf{agent}(X) \wedge \mathsf{greeting}(X) \wedge \neg\mathsf{greeting}(X) \wedge \neg\mathsf{agent}(X) \mapsto \mathsf{FALSE}$;
4. The Alternation rhetorical relation is ruled out as not holding between utterances π_{11} and π_{12}: $(\Sigma_{\mathsf{ALT}}(K(\pi_{11}), K(\pi_{12})) \mapsto \mathsf{FALSE}) \Rightarrow \neg\mathsf{ALT}(\pi_{11}, \pi_{12})$.

In fact, as it can be seen in figure 4, the Background rhetorical relation is found to hold between utterances π_{11} and π_{12}. The computations are performed, for any rhetorical relation computation, in the manner shown above; for further details, refer to [7], [8].

The discourse structure thus computed can be used, in utterance generation for dialogue, in order to trigger decisions concerning the surface form of machine speech turns, such as pronominal anaphora generation [5], semantic ellipsis control, or illocutionary force computation.

4 CONCLUSION

In this paper we have presented a domain-independent, general-purpose ontology of discourse features; this knowledge base, enforced with several discourse-related but task-independent axioms, can be used in human-computer dialogue systems, either for utterance interpretation or for generation. The ontology has a dual structure: the physical, XML form is doubled by a set of axioms, specifying either hierarchical relations, or discourse-specific, compositional relations.

The usage of the discourse ontology has been illustrated in rhetorical structuring for a book reservation human-computer dialogue task, showing how entities in the ontology, along with the axioms situating them, are coupled in order to define semantics for rhetorical relations; these semantics are checked in order to select rhetorical relations that hold between (logic forms of) utterances.

Yet, further usages of this ontology are envisageable, such as domain-specific information retrieval: indeed, textual productions can be searched in repositories according to particularities of the discourse involved, e.g., types of rhetorical relations, discourse structure patterns or topics involved.

ACKNOWLEDGEMENTS

We would like to thank Anne Xuereb from Cap Gémini, Grenoble for fruitful discussions concerning the design of the discourse ontology.

REFERENCES

[1] N. Asher and Alex Lascarides, *Logics of Conversation*, Cambridge University Press, United Knigdom, 2003.
[2] J. Caelen and A. Xuereb, *Interaction et pragmatique - jeux de dialogue et de langage*, Editions Hermès, Lavoisier, Paris, France, 2007.
[3] P. Ciccarese, E. Wu, and T. Clark, 'An overview of the swan 1.0 discourse ontology', *Proceedings of the 16th International World Wide Web Conference, Banff, Alberta, Canada, 2007*, (2007).
[4] J. Niekrasz and M. Purver, *A Multimodal Discourse Ontology for Meeting Understanding*, LNCS 3869, Springer Verlag, Heidelberg, 2005.
[5] V. Popescu, 'Pronominal anaphora generation in human-computer dialogue, using a semantic-pragmatic approach', *Proceedings of the Doctoral Consortium at the 8th FoLLI EUROLAN Summer School, 2007, Iasi, Romania*, 114–121, (2007).
[6] V. Popescu, J. Caelen, and C. Burileanu, *Generic Architecture for Natural Language Generation in Spoken Human-Computer Dialogue*, 57–72, Advances in Spoken Language Technology, Romanian Academy Publishing House, Bucharest, Romania, 2007.
[7] V. Popescu, J. Caelen, and C. Burileanu, *Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue*, ...–..., LNAI ...: Text, Speech and Dialogue, Springer Verlag, Heidelberg, 2007.
[8] V. Popescu, J. Caelen, and C. Burileanu, 'Using speech acts in rhetorical structuring for natural language generation in human-computer dialogue', *Proceedings of the 8th ACL Workshop on Discourse and Dialogue SIGDial 2007, Antwerp, Belgium*, 243–246, (2007).
[9] V. Popescu, J. Caelen, and Ioana Hagiescu, 'Automatic task ontology creation for human-computer dialogue, using word net and dialogue corpora', *Poster Session at the 4th IEEE Conference Speech Technology and Human-Computer Dialogue SpeD 2007, Iasi, Romania*, (2007).
[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey, USA, 2nd edn., 2003.