Syntax, Semantics and Pragmatics in Communication

Christoph Brandt
Université du Luxembourg & Tudor Luxembourg
SECAN-Lab & Living Lab
6, rue R. Coudenhove-Kalergi
L-1359 Luxembourg-Kirchberg
christoph.brandt@uni.lu

Marcus Kracht
Universität Bielefeld
Computational Linguistics and Mathematical Linguistics
C5-238, Postfach 10 01 31
33501 Bielefeld, Germany
marcus.kracht@uni-bielefeld.de

ABSTRACT

We develop a conceptual and formal understanding of syntax, semantics and pragmatics in communication from the point of view of (theoretical) computer science as well as (computational) linguistics. We motivate our understanding with the help of a real-world application scenario, apply it to a concrete example related to (business) dialogues in service oriented landscapes and finally point out how our results are related to the Semantic Web ecosystem in a way that is providing basic concepts for an emerging Pragmatic Web.

Keywords

syntax, semantics, pragmatics, protocols, communication

Categories and Subject Descriptors

D.3.1 [**Formal Definitions and Theory**]; E.4 [**Coding and Information Theory**]: Formal models of communication; F.3.2 [**Semantics of Programming Languages**]: Algebraic approaches to semantics, Denotational semantics, Operational semantics; F.4.3 [**Formal Languages Operations**]: Operations on languages

General Terms

Computational Linguistics, Pragmatics in Communication

1. INTRODUCTION

We present our understanding of syntax, semantics and pragmatics in communication. We reflect these notions from the point of view of (computational) linguistics [22] and (theoretical) computer science. After that we use our understanding to model (natural) dialogues in a service oriented landscape located in a business universe. This understanding was inspired by real-world requirements from Credit Suisse. Finally, we give our conclusion of how these findings could contribute to the Semantic Web ecosystem as well as an emerging Pragmatic Web.

The research question was to clarify what syntax, semantics and pragmatics in communication related to (business) dialogues might be. We look at this question from a computer science and linguistic point of view as well as from a conceptual and formal perspective. That way we reduce the broad notion of pragmatics in order to end up with a minimal core. We expect to add possible extensions later.

The contribution consists of a new and integrated understanding that is shown to be operational, as will be demonstrated with the help of a concrete example.

The paper is organized as follows: In the second section, we introduce a real-world application scenario to motivate our research. In the third section, we present our conceptual understanding of syntax, semantics and pragmatics in communication in order to prepare for the theoretical model, which follows in the fourth section. Afterwards, we apply our model towards a concrete example. We end with some conclusions, a discussion of future work as well as selected related work.

2. APPLICATION SCENARIO

The application scenario we started off with originates from a real-world situation at Credit Suisse. In detail, it was about how to handle security issues in IT landscapes which show up in modern SOA (service-oriented architectures) landscapes. However, by looking deeper into the scenario, we discovered that from a business perspective of the bank's priority should be on analyzing the business layer above the SOA landscape. This is because the security issues the bank is made accountable for are all derived from that layer. However, this layer was neither fully modeled nor formalized.

From a formal point of view, we can model a SOA landscape by the help of an ABT/Reo (Abstract Behavior Types) model as introduced in [2]. This enables us to talk about service oriented landscapes in an abstract way. For the sake of simplicity, we propose to use the ABT/Reo model likewise for the business universe, which facilitates a sound alignment later on, as it was shown in [8]. An ABT/Reo model is able to describe technical dialogues between service nodes, and it can, in principle, provide the same support for business dialogues. Fig. 1 shows how a very simple service oriented (business) landscape could look like.

The drafted scenario covers two persons (C,D) and two banks (A,B) as well as their mutual communication channels. The underlying idea is to provide a business service

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

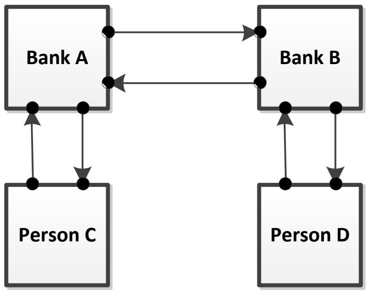

Figure 1: The drafted scenario

landscape that enables a person to transfer his or her own money to the account of someone else at another bank by the help of (natural) business dialogues.

However, doing this we were running into a couple of problems. Even though the ABT model is suitable to model checking [21] and ideally matches modeling requirements by supporting exogenous coordination and being compositional, it is unlikely to be used for the purpose of modeling (natural) business dialogues. People in the field simply do not want to touch formal methods. They would prefer to represent their business transactions in a naturally appearing business language that leads to readable business dialogues that are used to realize and document business transactions.

Therefore, it looks reasonable to take a linguistic point of view in order to complement computer science methods by linguistic methods that are closer to the concepts of the scenario. We believe that this is true because linguistic methods show good potential to be more suitable to the world of (natural) business transaction languages than computer science methods usually are. By doing this, we end up looking at reconstructed [28] DLs (domain language) that are enhanced by a communication aspect for the purpose of modeling business dialogues. In linguistics, such a DL would have to be specified regarding its syntax, semantics and pragmatics.

This approach faces certain problems, however. First, if computer science methods are to be complemented by linguistic methods, we need to conceptually clarify the notions of syntax, semantics and pragmatics from each perspective as well as their interplay. Second, the conceptual understanding needs to be formalized in order to make it operational. Third, we need to show that the formalized notion can be applied appropriately towards a concrete example.

3. CONCEPTUAL UNDERSTANDING

We will approach syntax, semantics and pragmatics first from the practical point of view by looking at DLs. Afterwards, we take a more theoretical stance by explaining how these notions fit into the different theoretical worlds of computer science and linguistics.

In practice, DLs are used in enterprise modeling to build organizational models [8]. However, we can imagine using them likewise for the purpose of representing (natural) business dialogues. In both cases, their syntax needs to be specified. Sometimes, their semantics is defined. By putting the use of DLs in the context of communication scenarios they are enriched, for example, by speech acts, protocols and communication policies. We would like to call this their communication aspect. These elements help to model the operational impact, following certain rules, a DL has when being used in communication, which we have identified to be the pragmatic dimension of a DL. However, DLs that are used in modeling scenarios usually do not have such an aspect and DLs in communication scenarios usually do not realize this aspect as an orthogonal component. However, when we assume that DLs are evolving over time, we run into potential change management problems because changes related to the communication aspect cannot be handled orthogonal to changes of non-communication aspects. As a consequence, changes in one part will cause the other part to break. In order to facilitate change management of DLs that are enriched by a communication aspect we would like to model the pragmatic dimension as an orthogonal component of the language definition of a DL. Secondly, we would like to do this in a way that enables us to easily extend a DL that comes without a communication aspect. That way we can hope to work with generic language models in the future that can be appropriately extended and configured.

In theory, theoretical computer science defines a DL by its syntax and semantics. The communication aspect is mapped towards these two concepts in an integrated way with its non-communication aspects. In order to end up with extensible language models, we propose to anchor the specification of the communication aspect at the meta-level. That would enable us to easily extend an existing DL without a communication aspect. Following this suggestion, a DL without a communication aspect can be specified at the object-level, its speech acts, protocols and possible communication policies can be defined at the meta-level. As a consequence, our extended DL would be a two-level-language encompassing an object- and meta-level at the same time. However, that is just the first step. The second step is about comparing the linguistic concept of pragmatics with the computer science concept of semantics. Here, we suggest to go with an internal (i) and an external (e) state. The semantics of a DL without the communication aspect would be realized by the external state, the semantics of the communication aspect by the internal state. Therefore, an external state is the state of the world and an internal state is the state of the dialogue between agents. We propose to assume that the protocol can have side-effects that lead to changes of the external state. As a consequence, the linguistic concept of pragmatics is realized by the computer science concept of syntax at the meta-level as well as the computer science concept of semantics at the meta-level by the help of an internal state. The assumed side-effects make it clear that, in practice, we have to deal with the concept of dependence between the internal and the external state that is causing quite some analytical problems in empirical studies. The following Fig 2 illustrates our conceptual understanding.

Therefore, pragmatics has two meanings: (a) a linguistic phenomenon, where the use of a language in dialogues is governed by protocols and policies, changing the state of a dialogue and therefore governing the next action in the communication and (b) an extralinguistic phenomenon, where the use of a language in ongoing dialogues is causing real-world states to change.

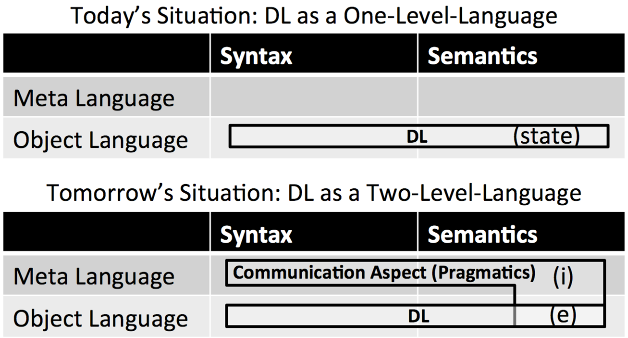

Figure 2: DLs Today and Tomorrow

4. THEORETICAL MODEL

In this section, we introduce the formal linguistic notions that complement the computer science notions related to the ABT/Reo model. In detail, we give definitions for channels, networks and messages before defining dialogues and protocols and closing by specifying syntax, semantics and pragmatics in communication.

4.1 Channels, Networks and Messages

Agents communicate by sending each other messages. Messages are sent around using channels, which distribute the physical messages. Messages must be adressed when sent. The address may consist of one or more recipients, as in emails, which can be sent not only to a single addressee but also to a list, either specified directly or using a mail alias. To model this we begin with what we call a communication structure: it specifies who can talk to whom via what channel, and also allows for using predicates for addresses (see [29]). The communication structure stays fixed over time, so time dependent behaviour with respect to access to communication is left out of consideration here. In the next definition, $\mathcal{P}(A)$ denotes the powerset of A.

Definition 1 (Communication Structure). *A* communication structure *is a quadruple* $(A, \mathcal{C}, P, g)$ *where* A *is a set of* agents*,* $\mathcal{C} \subseteq \mathcal{P}(A)$ *as set of* channels*,* P *a set of predicates and* $g\colon P \to \mathcal{P}(A)$ *an interpretation function.*

In an assumed real-world application scenario, the agents $A := \{\text{BankA}, \text{BankB}, \text{C1}, \ldots, \text{C6}\}$ are two banks and six customers. For the channels, we assume that C1 and C2 use secure online banking access to their banks BankA and BankB, respectively, modelled by channels $\{\text{C1}, \text{BankA}\}$ and $\{\text{C2}, \text{BankB}\}$. The other customers use a counter at BankA, where third persons are possible non-intended recipients of the conversations, modelled by a channel $\{\text{C3}, \ldots, \text{C6}, \text{Bank A}\}$. Moreover, the banks have a secure channel $\{\text{BankA}, \text{Ban kB}\}$ for inter-bank communication.

We will simply use one predicate for each agent in our system, i. e., $P := \{\texttt{bA}, \texttt{bB}, \texttt{c1}, \ldots, \texttt{c6}\}$ and $g(\texttt{bA}) := \{\text{BankA}\}$, $\ldots$, $g(\texttt{c6}) := \{\text{C6}\}$.

Based on the primitive predicates for agents we define agent selectors, which are complex predicates allowing to fine tune the list of recipients of a message.

Definition 2 (Agent Selector). *The set* $\mathrm{AS}(P)$ *of all* agent selectors *is recursively defined by* $\mathrm{AS}(P) := P \uplus \{\neg e, (e \wedge e'), (e \vee e') \colon e, e' \in \mathrm{AS}(P)\}$*. The* satisfaction *of an agent selector* $e \in \mathrm{AS}(P)$ *by an agent* $a \in A$*, written as* $a \models e$*, is recursively defined by* $a \models p \in P$ *iff* $a \in g(p)$*,* $a \models \neg e$ *iff* $a \not\models e$*,* $a \models (e \wedge e')$ *iff* $a \models e$ *and* $a \models e'$*, and* $a \models (e \vee e')$ *iff* $a \models e$ *or* $a \models e'$*. We extend the interpretation function* g *to a function* $\hat{g}\colon \mathrm{AS}(P) \to \mathcal{P}(A)$ *defined on all agent selectors by* $\hat{g}(e) := \{a \in A \colon a \models e\}$*. A communication structure is called* full *iff there is an agent selector* $\mathrm{e}_a \in \mathrm{AS}(P)$ *for each agent* $a \in A$ *such that* $\hat{g}(\mathrm{e}_a) = \{a\}$*.*

A message is an abstract entity consisting of a sentence, a sender and a recipient. When a message is actually sent, we have an utterance ([30], [19]). In contrast to messages, utterances are concrete, and they contain a time stamp and a channel. The channel specifies the physical recipients of route the message, while the address specifies the set of people that the message is intended for. Note that the same message can be sent several times over, thus creating several utterances.

Definition 3 (Message and Utterance). *A* message $m = (a, e, s)$ *consists of a* sender $a \in A$*, an* address $e \in \mathrm{AS}(P)$ *and a* sentence s *(see below). An* utterance $u = (t, C, m)$ *consists of a* timestamp t*, a channel* $C \in \mathcal{C}$ *and a message* m*. We assume timestamps to be totally ordered by a relation* $\leq$ *and extend this order to utterances by* $(t, C, m) \leq (t', C', m')$ *iff* $t \leq t'$*. The* recipients *of an utterance* $(t, C, (a, e, s))$ *are all agents* $b \in C \setminus \{a\}$ *and an agent* b *is* ratified *for a message* (a, e, s) *iff* $b \in \hat{g}(e)$*. An utterance* $u = (t, C, (a, e, s))$ *is* well-addressed *iff* $a \in C$ *and* $\hat{g}(e) \subseteq C$*, i. e., the sender and all ratified agents have access to the channel.*

In an assumed application scenario, possible utterances are (2011-02-28 11:00, {C1, BankA}, (BankA, bA, "Transfer €100 to C3's account at BankB.")) or (2011-02-28 10:00, {C3, ..., C6, BankB}, (C3, bB, "What is the balance of my account?")).

The model already allows us to precisely discuss the relations between the intended recipients, given in the address of a message, and the actual recipients on the channel of a concrete utterance. *Ratified recipients* are those agents to which the message is addressed and actually receive it, *non-ratified recipients* are those that receive it without being addressed, e. g., bystanders of a conversation in public. We have excluded *ratified non-recipients* for well-addressed utterances; finally, *non-ratified non-recipients* are all others ([24]). Addressing evidently is a crucial component in data security. For sensitive information it is to be avoided at all costs to have bystanders. This creates interesting complications for public announcement logic ([5]) that are worthy of further study.

4.2 Dialogues and Protocols

Dialogues are sequences of utterances. Thus they are concrete entities. From the standpoint of the participants, the dialogue is not structured enough. What they really need to know is not only who said what when, but also who said what in reply to what. This is known as the *turn structure*; it identifies certain utterances as responses to others. Though often it is given explicitly (as in the header of an email), often enough there is no such annotation.

Definition 4 (Dialogue). *A* dialogue Δ *is a set of utterances, which is implicitly ordered due to the ordering on the timestamps. A* structured dialogue (Δ, τ) *is a dialogue* *with a* turn structure $\tau \subseteq \mathcal{P}(\Delta)$ *such that all turns* $t \in \tau$ *are pairs* $|t| = 2$. *A structured dialogue is* nesting *iff there are no* $\{u, u'\}, \{v, v'\} \in \tau$ *with* $u \leq v \leq u' \leq v'$.

The turn structure in human 2-person dialogues is typically but not universally nesting (see [15], [3]). There are notable exceptions to the nesting structure, and humans are also capable of handling dialogues in an open and flexible way. *Thus is it is in general not advisable to simply transfer existing protocols for machines for the study of interaction with (and between) humans.*

For example, suppose in u_1, C1 orders his bank to transfer € 100 to C3's account. Next, the bank asks which account of C1 is to be used to transfer from (u_2). After C1 has specified the account (u_3), the bank acknowledges the transfer (u_4). The linear ordering is $u_1 < u_2 < u_3 < u_4$, while the induced turn structure is $\{\{u_1, u_4\}, \{u_2, u_3\}\}$, since u_4 is in response to u_1, while u_3 is a response to u_2. Note that C1 must know that the bank is acknowledging u_1 via u_4, otherwise he would have to repeat the request.

Thus, for a dialogue to be transparent it must have a turn structure. If it is not explicitly given we must be able to recover it. This can be done with the help of well-formedness conditions on dialogues, called protocols.

DEFINITION 5 (PROTOCOL). *A* protocol Π *is a property of structured dialogues. A structured dialogue* (Δ, τ) fulfills Π *iff there exists* $\tau' \supseteq \tau$ *such that* (Δ, τ') *satisfies* Π. *A protocol is* stable *iff for each* (Δ, τ) *there is at most one* $\tau' \supseteq \tau$ *such that* (Δ, τ') *satisfies* Π.

Protocols are thus essential properties of dialogues. They are therefore enforced. While protocols specify nonviolable properties of dialogues, policies can in principle be violated. Policies are implemented for various reasons, for example to guarantee security, to be polite and so on. The main tool for analysing policies is modal logic, in particular deontic logic. This would also allow to verify compliance with policies by a machine. We sketch an approach, known as *Anderson-Kanger reduction* (see e. g. [18]). We begin with a set *cd* of so-called *compliant dialogues*. A *moment* is a pair (d, t) where d is a dialogue and t is an element of t. We write $(d, t) \vDash \Box\chi$ if for all d' such that $d' \in cd$ such that d' differs from d only in the part after t, $(d', t) \vDash \chi$.

DEFINITION 6 (POLICY). *A* policy *is a sentence of the form* $\varphi \to \Box\chi$ *where both* φ *and* χ *are free of* $\Box$. *It is satisfied by a dialogue* d *if for all members* t*: if* $(d, t) \vDash \varphi$ *then for all compliant variants* d' *of* d *at* t*:* $(d', t) \vDash \chi$.

Here, we think of the actual policy as χ, while φ is a 'guard'. Setting $\varphi := \top$ means that the policy is enforced everywhere. A more refined model can be built using an action based language such as dynamic logic or stit-theory (see [6]).

While the scheduling of dialogues is often nondeterministic in the human context, it is safe to assume that dialogues with machines proceed deterministically. That is, the corresponding machine will know at each point what kind of action it will have to perform in the dialogue.

Speech act types are essential in shaping dialogues. They are associated with specific protocols.

DEFINITION 7 (SENTENCE). *A* sentence (c, p) *consists of a* speech act type c *and an open proposition (see below)* p *as* content. *The possible speech act types are* Statement *(abbreviated as* $\vdash$*),* Question *(abbreviated as* **?***),* Command *(abbreviated as* **!***),* Promise *(abbreviated as* ☞*) and* Acknowledgement *(abbreviated as* ✓*).*

We can already give some requirements that are common to all protocols using only the speech act types. Namely, each question has to be answered by a statement and each command by an acknowledgement, formally for each $u = (t, C, (a, e, (\mathbf{?}, p))) \in \Delta$ and each $b \in \hat{g}(e)$ there has to be a $u' = (t', C', (b, e', (\vdash, p'))) \in \Delta$ with $u \leq u'$, $\{u, u'\} \in \tau$ and $a \in \hat{g}(e')$ and analogously for **!** and ✓. Thus, acknowledgements are not responses to questions, statements are not proper responses for promises and so on.

4.3 Items

From a linguistic point of view, language operates on three levels: syntax, semantics and pragmatics. Linguistic items are triples $\sigma = (e, m, p)$, where e specifies the syntax, m the semantics and p the pragmatics. In each of the three we have both an identification of the substantive properties (say, that the syntax of the item is a string) and the operative properties (say, that the string, being a transitive verb, is to be composed with its object to the right). We shall not have much to say about the syntax of items. Also, the semantics will be straightforward. Therefore, we concentrate instead on the pragmatics.

As concerns the division between semantics and pragmatics, we distinguish between *external states* and *internal states*. An external state is a state of the world. In our model, it is the set of accounts together with the information about the amounts as well as the account holders and banks. The internal states are 8-tuples, with one state for each agent. An utterance changes these states.

Notice that the semantics of certain items is fairly trivial. An acknowledgement such as (✓, `OK`) has only a trivial meaning. Its sole purpose is to close a turn. However, its pragmatics is not trivial. If some action is requested, we may refuse the request or we may honour it. (✓, `OK`) is a way to express that we are intent on honouring it.

Here is a simple example. The verb "to open" can be used in a command to open an account. To simplify even further, we assume that the syntax is /`Open!`/, which is used by the speaker to ask for an account to be opened by the recipient. This is a sentence (!, `Open`), where ! signals a command and /`Open`/ is the proposition expressing the content. Commands are either acknowledged (via the acknowledgment (✓, `OK`)) or refused.

The external states are the collections of account data. The data structure δ_i a bank keeps at any moment i is (1) a list of pairs (o, a) where a is account number and o its owner (one of p_1 through p_n), and (2) a list containing pairs (a, b), where a is the account number, and b the balance. To keep track of its own actions a bank may keep a record of the Δ_j for past moments ($j < i$). A person may also have cash. Thus, for each person we associate a number, which is the cash owned by that person.

Syntax: /`Open!`/ is an imperative sentence.

Semantics: For all $b \in g(e')$, the action $\mathsf{open}'(a)(b)$ is executed, where a is a person and b is a bank. Here, $\mathsf{open}'(a)(b)(s)$ is the following state $s' = (m_1, m_2, \Delta_1', \Delta_2')$. If $b = b_1$ then $\Delta_2' = \Delta_2$. Given $\Delta_1 = (\ell_1, \ell_2, \ell_3)$, we have $\Delta_1' = (\ell_1', \ell_2', \ell_3')$, where $\ell_1' = \ell_1$, $\ell_2' = \ell_2 \cup$

$\{(a, \vec{x})\}$ and $\ell_3' = \ell_3 \cup \{(i, 0)\}$, where i is the smallest number not assigned in ℓ_2. If $b = b_2$ then the result is analogous with 1 and 2 interchanged. (This action is deterministic; one need not do that but it is simpler.)

Pragmatics: If Δ contains $(t, C, (a, e_b, (!, \texttt{Open})))$ then τ contains $\{u, u'\}$ such that $u' = (t', \{a, b\}, (b, e_a, (\vdash, \texttt{I}_\sqcup \texttt{opened}_\sqcup \vec{x})))$ or $u' = (t', \{a, b\}, (b, e_a, (\checkmark, \texttt{Sorry})))$ where $t' > t$. (Issuing the response to recipients other than a is a violation of protocol. Using a channel other than $\{a, b\}$ in the response is likewise in violation of protocol. Also, no bank may open an account without being requested to do so.)

We have specified the pragmatics here as conditions on the dialogue (in other words, as a protocol). To understand how the participants can keep the protocol we have to define internal states. Each agent a has an internal state $I(a)$, which consists of the following: (a) a dialogue $\Delta(a)$ (which is a record of all transactions that a witnessed) together with a set τ of identified turns, (b) a stack $S(a)$ of tasks (aka TODO-list), (c) a stack $T(a)$ of requests issued to other agents. Suppose, there is an utterance u containing a message addressed to a, and a receives it. Then either case may arise.

1. u closes a turn that a opened; in this case, a removes the top from $T(a)$, say v, provided v matches u. Otherwise, the protocol is not kept.

2. u opens a turn by requesting something from a. u is put on top of $S(a)$.

In any case the utterance is added to $\Delta(a)$. Agents constantly try to empty the stack $S(a)$. a can eliminate entries from $S(a)$ by performing the described action, for example by closing the turn appropriately. Each time a closes a turn, the resulting utterance pair is added to τ. Notice that the dialogues $\Delta(a)$ are redundant for the purpose of scheduling messages. Also, $\Delta(a)$ is a reduct of Δ, containing all messages that a has received or sent.

The nestedness of turn structures is due to the fact that requests are kept in a stack, so that the last request received is dealt with first. (This assumes actually a 2 person dialogue. If more parties are involved, a separate stack $S(a, b)$ needs to be kept for each other agent b.) Also, the schedule of tasks is treated on a last-in-first-out basis.

Notice that an utterance may give rise to much more than a simple turn and thus create complex actions on $Q(a)$. A transfer of money, for example, involves several transactions in a row. Upon the request by a to transfer money, the bank will have to verify the data with the receiving institution, send the money to that institution, and, upon acknowledgment, acknowledge in turn to a. *This means that the factual structure of the dialogue is heavily dependent on the semantics. Thus, semantics and pragmatics are interdependent in a nontrivial way.*

5. A CONCRETE EXAMPLE

We provide a concrete example that shows how our formal model can be applied to a very simplified real-world case.

In detail, a mother (m) wants to transfer € 200 to the account of her daughter (d). Let the agents be $\{m, d, b, c\}$, where b is the bank of m, c the bank of d. The predicates are M, D, B, C, with obvious interpretation. Assume as channels $\{\{m, d\}, \{m, b\}, \{b, c\}, \{d, c\}\}$. External states consist of data on cash for each individual, data on account holders and balance for each bank. The initial external state is

Cash

m :	250
d :	25

B

Name	Acct
M	136

Acct	Bal
136	175

C

Name	Acct
D	71

Acct	Bal
71	13

(1)

Internal states consist in a series of stacks for each person: a todo-list, and a set of message stacks. Initially, the stacks are empty. Since the account 136 does not have enough funds, m pays from her cash € 75 into the account. This changes the state into

Cash

m :	*175*
d :	25

B

Name	Acct
M	136

Acct	Bal
136	*250*

C

Name	Acct
D	71

Acct	Bal
71	13

(2)

Now consider the following utterance:

$$(t, \{m, b\}, (m, \texttt{B}, (!, /\texttt{Transfer € 200 from account 136 to account 71 at bank C.}/))) \quad (3)$$

Once issued, the request changes the stack of bank B. The bank will now push a money transfer on the todo-list, and the utterance on the message stack (since it has not been acknowledged). The transfer item triggers a series of actions: (a) check whether the account 136 has enough funds, (b) check with the bank C whether account 71 exists, and (c) ordering C to change the balance of the account. Upon acknowledgement by C of the requested action, B will deduct the amount from account 136, and, finally, send an acknowledgement:

$$(t', \{m, b\}, (b, \texttt{M}, (\checkmark, /\texttt{OK}/))) \quad (4)$$

At this point all stacks are cleared.

We may note that first transferring € 200 and then paying € 75 into the account does not have the same effect. The transfer is rejected on the grounds that the funds are not sufficient. The bank will then refuse the transaction and the external state remains unchanged.

$$(t', \{m, b\}, (b, \texttt{M}, (\checkmark, /\texttt{Fail}/))) \quad (5)$$

For the purpose of the next definition let $p_1 : E \times I \to E$ be the first projection function, i. e. $p_1(e, i) = e$.

Definition 8 (Semantic Equivalence). *Let $f, g : E \times I \to E \times I$ be functions on the state set. We say that f and g are* weakly semantically equivalent *if for all (e, i): if $f(e, i)$ and $g(e, i)$ are both defined then $p_1(f(e, i)) = p_1(g(e, i))$. We say that f and g are* strongly semantically equivalent *if for all (e, i): (a) $f(e, i)$ is defined iff $g(e, i)$ is defined, and (b) if $f(e, i)$ is defined then $p_1(f(e, i)) = p_1(g(e, i))$.*

Let f be the compound action of paying into the account and then transferring, and g the compound action of transferring before paying into the account. Then f and g are weakly semantically equivalent: if both actions succeed they reach the same (external) state. But they are not strongly equivalent. It may e. g. happen that g is undefined while f is defined. Such is the case, for example, when g accidentally violates some protocol properties, e. g. when issuing a second transfer before the first one was acknowledged, provided this action sequence is actually excluded by the protocol.

The above definition assumes that dialogues can be aborted in cases of error. However, it is preferrable to have explicit rules to handle errors that arise in either the semantic or the pragmatic domain. This would require the introduction of specific error states and an appropriate change of the definitions.

Pragmatic equivalence is defined similarly, with the second projection p_2 replacing p_1. Transferring some amount to one person is pragmatically equivalent to transferring it to another person, since eventually the internal states are balanced. However, the semantic difference is noticeable.

On the other hand there are examples of actions which are semantically but not pragmatically equivalent. One such action is sending out information. Information does not change the balance sheets, hence the external states remain. But it requires an acknowledgement of the parties receiving it.

Note also the concept of independence, as discussed in [23]. Consider a (partial) function $f : E \times I \to E \times I$. It can be seen as a pair of functions $f(e, i) = (g(e, i), h(e, i))$ where $g : E \times I \to E$ and $h : E \times I \to I$. This functions may show partial or total neglect for some of their arguments. Thus, $f : E \times I \to E \times I$ is *independent* if there are partial functions $g : E \to E$ and $h : I \to I$ such that $f(e, i)$ is defined iff $g(e)$ and $h(i)$ are defined, and in that case $f(e, i) = (g(e), h(i))$. If f is not independent, the semantic and pragmatic levels are said to *interact*. Interaction can be asymmetric. For example, if $f(e, i) = (g'(e, i), h(i))$ for some partial $g' : E \times I \to E$ then semantics depends on pragmatics; similarly, if $f(e, i) = (g(e), h'(e, i))$ for some $h : E \times I \to I$. In a development scenario, where users actually implement e. g. policies it is interesting to track the dependencies between the levels both of semantics and pragmatics. This would allow to monitor the effect of particular changes made to the semantics and/or pragmatics.

6. RELATED WORK

The literature on this subject is broad and heterogeneous. We therefore present only selected references in order to put our understanding of pragmatics in communication into relation with the understanding of others from the area of computer science and linguistics.

6.1 Computer Science

In [1], an agent communication language (ACL) is presented using interaction protocols and policies that contextually enrich the minimal meaning of speech acts. The authors aim to introduce a framework with a semantics that consists of a complete catalogue of communicative actions encompassing a complementary pragmatic component that accounts for the social effects of performing a communicative action and thereby facilitates the achievement of its perlocutionary effects. However, their understanding does not lead to extensible language models.

In [12, 13], the Pragmatic Web is defined by extending the Semantic Web with the notion of context for Semantic (Web) resources as well as the notion of context for information use. This understanding is orthogonal to the one presented in this paper because it does not explain how protocols and communication policies can be used to explain pragmatics in communication.

In [20], agent communication languages are identified to be relevant in e-commerce and industry. It is further said that such a communication language needs a precisely defined syntax, semantics and pragmatics that provides the basis for communication between independently designed and developed software agents. This confirms the result of the presented analysis in this paper. However, it does not cover business dialogues in reconstructed domain languages [28] as in our scenario. It further abstracts away from human-to-machine communication scenarios.

In [17], the authors claim that an intersection-based matchmaking is insufficient to ensure applicability of web services for a given request. They further claim that, in order to fix this problem, pragmatics of web services need to be defined. They show that although most of prior approaches providing a formal semantics for web services, their pragmatics to describe requests is improper since it differs from the user intention. In order to provide a practical discovery approach, the authors clearly specify the intention and the interpretation of preconditions and effects. This approach has some similarities to the one presented in this paper. However, it still sticks to a purely technical communication scenario.

In [25], the authors present STAIRS as a method for the compositional development of interactions in the setting of UML 2.0. In addition to defining denotational trace semantics for the main aspects of interactions, STAIRS is particularly concerned with refinement, and the authors have given guidelines on how to refine interactions by adding behaviours (supplementing), removing underspecification (narrowing) or by decomposition (detailing). Here, the authors focus on the modeling process, not on generic language models encompassing pragmatics in communication.

In [27], the authors present a story board approach, which in an abstract way specifies who will be using a system, in which way and for what goals. They claim that while the syntax and semantics of story-boarding has been well explored, its pragmatics has not. They further claim that both user models and life cases define the core story board pragmatics. In addition to that, they say a deeper understanding of contexts is needed, which is left for future work. However, their approach remains semi-formal, in contrast to the results presented in this paper.

6.2 Linguistics

In [16], knowledge of conversational pragmatic structure was examined by asking 53 female volunteers to rate the naturalness of three versions of an appointment-making conversation from a beauty salon. The data was used as evidence to promote the idea that conversationalists are motivated agents pursuing goals in communication by engaging in actions. The data can be used to study pragmatics as an empirical phenomenon and therefore complements the conceptual, formal and normative approach presented in this paper.

[10] highlights that there is still little consensus on how semantics and pragmatics can be properly differentiated. It proposes the following clarification, quoting [4]: "*Pragmatic information is (extralinguistic) information that arises from an actual act of utterance. Whereas semantic information is encoded in what is uttered, pragmatic information is generated by, or at least made relevant by, the act of uttering it*", and further: "*Taken as properties of sentences, semantic*

properties are on a par with syntactic and phonological properties: they are linguistic properties. Pragmatic properties, on the other hand, belong to acts of uttering sentences in the course of communicating. Sentences have the properties they have independently of anybody's act of uttering them." This understanding supports the one presented in this paper, however, it is not yet formalized.

In [9], the NXT-format Switchboard Corpus represents a long-standing corpus of telephone conversations. It is a rich resource for potential investigations of linguistic features that can be found in these dialogues and their concrete interactions. This empirical approach complements the normative approach presented in this paper.

In [26], it is said that normative pragmatics calls attention to how the manifest strategic design of a message produces interpretive effects and interactional consequences. The authors claim that the real focus of argumentation studies is on the messages people produce as they try to decide what to believe, how to feel, and what to do. Therefore, finding a theoretically productive way to come to grips with real messages in real controversies has been the most important theme of recent argumentation study. This supports our choice to work with internal and external states from an empirical point of view.

7. CONCLUSION AND FUTURE WORK

The main conclusion is that, from the point of view of linguistics, pragmatics in communication can be modeled with the help of internal and external states. This allows us to talk about semantic and pragmatic equivalence in an explicit way. It further enables us to look deeper into the interplay of protocol issues and the semantics of a language as brought up by Clark in [11]. As a consequence, it clears the way of how to model pragmatic aspects as an orthogonal component, which facilitates the change management of semantic and pragmatic issues. We believe that the last point can be of practical relevance in the examined enterprise modeling scenarios.

Since the techniques and definitions used here can be naturally extended to graph based formalisms [14] that are highly relevant in the area of enterprise modeling, as it can be seen in [8], possible future work shows up in the area of dialogue-oriented modeling. To do this, the presented components simply need to be adapted, e. g. by changing syntactic structures from strings to graphs. Nothing of substance changes.

Dialogue-oriented modelling shows good potential to change the process of building enterprise models. Today, enterprise models are created by users using tools. Tomorrow, they could be built in the context of a dialogue between a modeling system and a modeling agent. Here, pragmatics in communication shows up in the way speech acts and modeling protocols as well as modeling policies are used to build enterprise models and are applied to manage their underlying DLs. Doing all this with the help of open and flexible dialogues between a technical and a human agent is ideally supporting agile modeling techniques. In addition to that, there is hope to use the proposed formal model in order to describe online banking dialogues using natural language protocols that are almost non-extisting in such scenarios.

By putting DLs and enterprise models on top of RDF graphs [7] the Semantic Web ecosystem can be used for the purpose of dialogue-oriented enterprise modeling. We believe that this will lead to questions of how the Semantic Web can be extended by a communication aspect, or being more concrete, if heterogeneous and distributed web-based communication based on certain protocols could be a way to go to provide a sound understanding of the notion of a Pragmatic Web. We hope that the presented understanding of pragmatics in communication will help to clarify the theoretical foundations needed here. However, even this is looking very promising, it still requires substantial further research.

8. REFERENCES

[1] R. Agerri and E. Alonso. A semantic and pragmatic framework for the specification of agent communication languages: Motivational attitudes and norms. In M. Kolp, P. Bresciani, B. Henderson-Sellers, and M. Winikoff, editors, *AOIS*, volume 3529 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2005.

[2] F. Arbab. Abstract Behavior Types: A Foundation Model for Components and Their Composition. In F. S. de Boer, M. M. Bonsangue, S. Graf, and W.-P. de Roever, editors, *Formal Methods for Components and Objects*, volume 2852 of *Lecture Notes in Computer Science*, chapter 2, pages 33–70. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2003.

[3] N. Asher. Varieties of Discourse Structure in Dialogues. In *Proceedings of the Second Workshop on the Semantics and Pragmatics of Dialogue (Twendial '98)*, 1998.

[4] K. Bach. You don't say? *Synthese*, 128:15–44, 2001.

[5] P. Balbiani, A. Baltag, H. van Ditmarsch, A. Herzig, T. Hosi, and S. de Lima. 'Knowable' as 'Known After An Announcement'. *The Review of Symbolic Logic*, 1:305–334, 2008.

[6] M. M. Bentzen. *Stit, Iit, and Deontic Logic for Action Types.* PhD thesis, Department of Philosophy, Roskilde University, 2010.

[7] B. Braatz and C. Brandt. Domain-specific modelling languages with algebraic graph transformations on RDF. In B. A. Malloy, S. Staab, and M. van den Brand, editors, *SLE*, volume 6563 of *Lecture Notes in Computer Science*, pages 82–101. Springer, 2010.

[8] C. Brandt and F. Hermann. How far can enterprise modeling for banking be supported by graph transformation? In H. Ehrig, A. Rensink, G. Rozenberg, and A. Schürr, editors, *ICGT*, volume 6372 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2010.

[9] S. Calhoun, J. Carletta, J. Brenier, N. Mayo, D. Jurafsky, M. Steedman, and D. Beaver. The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44:387–419, 2010.

[10] R. Carston. Linguistic communication and the semantics/pragmatics distinction. *Synthese*, 165(3):321–345, 2008.

[11] H. H. Clark. *Using Language.* Cambridge Universtiy Press., 1996.

[12] A. de Moor. Patterns for the pragmatic web. In F. Dau, M.-L. Mugnier, and G. Stumme, editors, *ICCS*, volume 3596 of *Lecture Notes in Computer*

Science, pages 1–18. Springer, 2005.

[13] A. de Moor, M. Keeler, and G. Richmond. Towards a pragmatic web. In U. Priss, D. Corbett, and G. Angelova, editors, *ICCS*, volume 2393 of *Lecture Notes in Computer Science*, pages 235–249. Springer, 2002.

[14] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theoretical Computer Science. Springer Verlag, 2006.

[15] R. Fernandez and U. Endriss. Abstract models for dialogue protocols. *Journal of Logic, Language and Information*, 16:121–140, 2007.

[16] D. Goldthwaite. Knowledge of pragmatic conversational structure. *Journal of Psycholinguistic Research*, 26:497–508(12), September 1997.

[17] M. Junghans, S. Agarwal, and R. Studer. Towards practical semantic web service discovery. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, editors, *ESWC (2)*, volume 6089 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2010.

[18] S. Kanger. New Foundations for Ethical Theory. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 36–58. Reidel, Dordrecht, 1971.

[19] S. Kimbrough and M. Thornburg. On semantically-accessible messaging in an office environment. In *Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences. Vol.III: Decision Support and Knowledge Based Systems Track*, volume 3, pages 566 – 574, 1989.

[20] D. Kinny. Reliable agent communication - A pragmatic perspective. In *Proceedings of PRIMA 1999*, pages 16–31, 1999.

[21] S. Klüppelholz and C. Baier. Symbolic model checking for channel-based component connectors. *Sci. Comput. Program.*, 74:688–701, July 2009.

[22] M. Kracht. *The Mathematics of Language*. Studies in Generative Grammar 63. Mouton De Gruyter, Berlin, 2003.

[23] M. Kracht. *Interpreted Languages and Compositionality*. Studies in Linguistics and Philosophy. Springer, Heidelberg, 2012.

[24] J. McCawley. Participant Roles, Frames, and Speech Acts. *Linguistics and Philosophy*, 22:595–619, 1999.

[25] R. K. Runde, Ø. Haugen, and K. Stølen. The pragmatics of STAIRS. In F. S. de Boer, M. M. Bonsangue, S. Graf, and W. P. de Roever, editors, *FMCO*, volume 4111 of *Lecture Notes in Computer Science*, pages 88–114. Springer, 2005.

[26] J. S. Rhetoric and dialectic from the standpoint of normative pragmatics. *Argumentation*, 14:261–286(26), August 2000.

[27] K.-D. Schewe and B. Thalheim. User models: A contribution to pragmatics of web information systems design. In K. Aberer, Z. Peng, E. A. Rundensteiner, Y. Zhang, and X. Li, editors, *WISE*, volume 4255 of *Lecture Notes in Computer Science*, pages 512–523. Springer, 2006.

[28] B. Schienmann. *Objektorientierter Fachentwurf: ein terminologiebasierter Ansatz für die Konstruktion von Anwendungssystemen*. Teubner-Texte zur Informatik, 1997.

[29] R. Sharp. *Principles of Protocol Design*. Springer, 2008.

[30] H. Weigand and W. Van Den Heuvel. Meta-patterns for electronic commerce transactions based on FLBC. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, volume 4, pages 261 – 270, 1998.