

BRINGING SPEECH ACTS INTO UMM

Maria Bergholtz, Prasad Jayaweera, Paul Johannesson and Petia Wohed

Department of Computer and Systems Sciences

Stockholm University and Royal Institute of Technology

Forum 100, SE-164 40 Kista

Stockholm, Sweden

{maria, prasad, pajo, petia}@dsv.su.se

1st Int. REA Technology Workshop Copenhagen, Denmark, April 22nd-24th, 2004

1 INTRODUCTION

In order to understand human communication, it is important to consider the means by which communication is carried out as well as the effects produced by communication. One way to do this is to view human actions as taking place in three different worlds:

- *The physical world.* In this world, people carry out physical actions – they utter sounds, wave their hands, send electronic messages, etc.
- *The communicative world.* In this world, people express their intentions and feelings. They tell other people what they know, and they try to influence the behaviour of other actors by communicating with them. People perform such communicative actions by performing actions in the physical world.
- *The social/institutional world.* In this world, people change the social and institutional relationships among them. For example, people become married or they acquire possession of property. People change social and institutional relationships by performing actions in the communicative world.

Examples

- I utter the words “I promise to do the dishes” (an action in the physical world). Thereby, I express my intention to do the dishes (an action in the communicative world). Thereby, a commitment for me to do the dishes comes into existence (a relationship between my family and me in the social/institutional world).
- A priest utters the words “I pronounce you husband and wife” (an action in the physical world). Thereby, he expresses his intention to wed two persons (an action in the communicative world). Thereby, a marriage, a sort of contract, is established (a relationship between two persons in the social/institutional world).

A main observation is that language can be used not only to describe the reality but also to change it. In other words, to speak is to act. Linguistic acts that intend to influence the reality are commonly called *speech acts*.

In the rest of this document, we will discuss how speech acts can be used for extending UMM. The main idea is that speech acts can function as a bridge between the “physical world” and the “social/institutional” world in UMM. By the “physical world” in UMM is meant services, messages, and transactions – essentially the BSV and BTV. By the “social/institutional” world in UMM is meant the economic elements in the BRV. (As UMM focuses on economic aspects, we will henceforth talk about the economic world instead of the more general social/institutional world.) Speech acts express what people want to do with economic elements (commitments, contracts, etc.). Thus, the speech acts motivate why

people carry out physical actions like sending messages. This allows for a nice design approach. First, we think about the economic world, i.e. what economic effects people want to achieve – these are things like establishing contracts and commitments, fulfilling the commitments, and so on. Secondly, we think about the communicative world, i.e. what people do in order to achieve these economic effects – they express their intentions by requesting, promising and declaring, etc. Thirdly, we think about the physical world, i.e. the physical actions people carry out in order to express their intentions – they send messages. Thinking in these three layers will make it easier to involve business people in analysis and design, as they can talk about things that are meaningful from a business perspective (people requesting and promising, etc.) and they do not need to talk about technical artifacts like state machines and messages. Furthermore, we can obtain good traceability from high-level business concepts down to low-level technology concepts.

The rest of the document is organized as follows. Section 2 describes in more detail the structure of speech acts as adapted to the UMM framework. Section 3 outlines the relationship between the physical world and the communicative world in the context of UMM, and Section 4 does the same for the economic and the communicative world. Section 5 sketches some applications and benefits of the speech act approach. Section 6 introduces a number of open issues and Section 7 concludes by briefly surveying some related work.

2 PRAGMATIC ACTIONS

In this section, we discuss how speech acts can be adapted to the UMM framework. We will use the term “pragmatic action” instead of speech act in order to emphasize that we make an adaptation to the UMM. Below, we also use the term “intention” instead of the more customary “illocutionary force”.

Definition: A *pragmatic action* is a triple <Intention, Action, REAObject>.

Intuitively, the components mean the following:

- REAObject is an object that you want to act upon in some way, an economic element from BRV
- Action is the action to be applied on the REAObject – create, change, or cancel it
- Intention tells what intention you have to the (Action on the REAObject)

Formally, we define Intention and Action through enumeration:

- Intention = {propose, accept, reject, declare, query, reply, assert}
- Action = {create, change, cancel, none}

Furthermore, a REAObject is an instance of one of the classes Agreement, EconomicContract, EconomicComitment, and EconomicEvent.

So, the intentions are the following:

- *propose* – someone proposes to create, change, or cancel an REAObject
- *accept* – someone accepts a previous proposal
- *reject* – someone rejects a previous proposal
- *declare* – someone unilaterally creates, changes, or cancels a REAObject
- *query* – someone asks for information

- *reply* – someone replies to a previous query
- *assert* – someone makes a statement about one or several REAObjects

For query, reply, and assert, there is no relevant Action involved, so only the “dummy” none can be used.

Examples:

- I want to create a contract. I propose (an Intention) to create (an Action) a contract (an REAObject).
- I cancel a commitment. I declare (an Intention) to cancel (an Action) a commitment (an REAObject).

So, a pragmatic action attempts to capture that someone expresses an intention towards some action on an REA object. This is a purely communicative act, someone expressing something to someone else. We also need to figure out how and when the communicative act happens, and this is the topic of the next section. In the section after that, we will discuss how the economic world is affected as a consequence of the pragmatic action.

3 PRAGMATIC ACTIONS AND UMM

In terms of the three domains introduced in Section 1, UMM explicitly addresses only the physical and the social/institutional domains. The physical domain is modeled through classes like *BusinessTransaction* and *BusinessAction*, while the social/institutional domain is modeled through REA concepts like, *EconomicCommitment*, *EconomicEvent*, and other classes. The details of the communicative domain, however, are not explicitly modeled. This state of affairs causes two main problems. First, the relationship between the physical and the social/institutional domains is very coarsely modeled; essentially the UMM only states that a completed collaboration may influence objects in the social/institutional world, but it does not tell how the components of a collaboration affect the social/institutional objects. Secondly, there is no structured or systematic way of specifying how events in the physical domain influence the social/institutional domain. These problems can be overcome by introducing the communicative domain as an additional layer in the UMM, thereby creating a bridge between the physical and social/institutional domains.

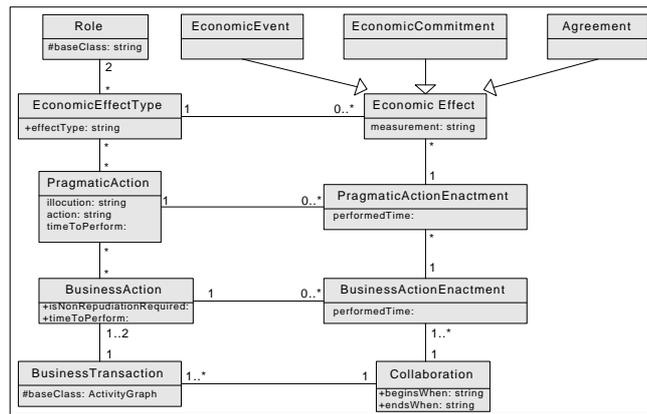


Fig. 1 Extension for REA concepts in BRV of UMM

As a preparation to modeling the communicative domain, a minor modification to UMM BRV is made, see Fig. 1. A class EconomicEffect is introduced as a superclass of EconomicCommitment, Agreement, and EconomicEvent.

The power type of EconomicEffect, called EconomicEffectType, is also added for the purpose of differentiating between the modeling of concrete, tangible objects in a domain, and the abstract characteristic categories of these objects.

These modifications will allow for a more concise representation of the effects (as well as the characteristics of the effects) of communicative actions. In addition to these changes, the classes BusinessActionEnactment and BusinessTransactionEnactment are added. These represent the actual execution of a business action or business transaction, respectively.

4 PRAGMATIC ACTIONS AND REA

When a pragmatic action occurs, it typically gives rise to effects in the “economic world” of UMM. Such an effect can be represented in many different ways, but one natural way seems to be to use the state machines of the business entities. This means that to each transition in such a state machine, we associate a pragmatic action. When that pragmatic action occurs, it will trigger the transition and move the state machine forwards. An example is given in Fig. 2 below, where a simple state machine for a contract fulfillment (contract formation has been excluded here) is given with the pragmatic actions as labels on the arcs. This is a part of entire contract formation and contract fulfillment state machine.

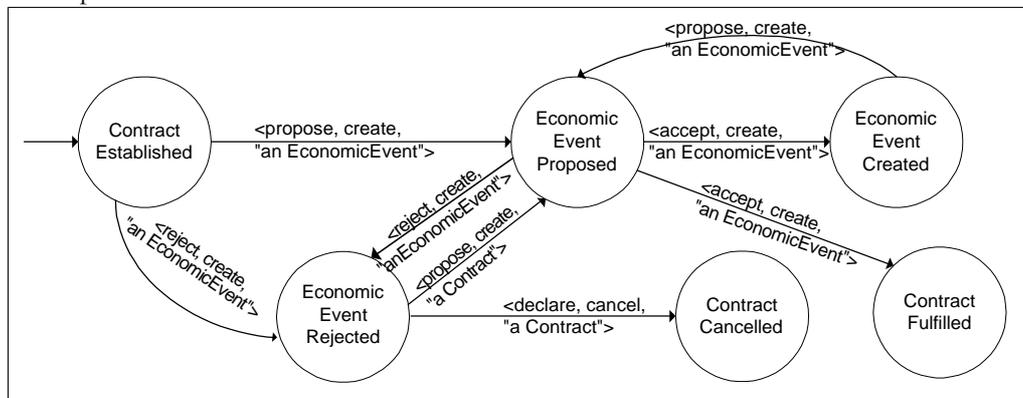


Fig. 2 State Machine for Contract Fulfillment

5.1 BASIC CONVERSATION FOR ACTION ANALYSIS

Basic Conversation for Action is a well-known example of an adapted application of Speech Act Theory. It was proposed by T. Winograd and F. Flores in 1986.

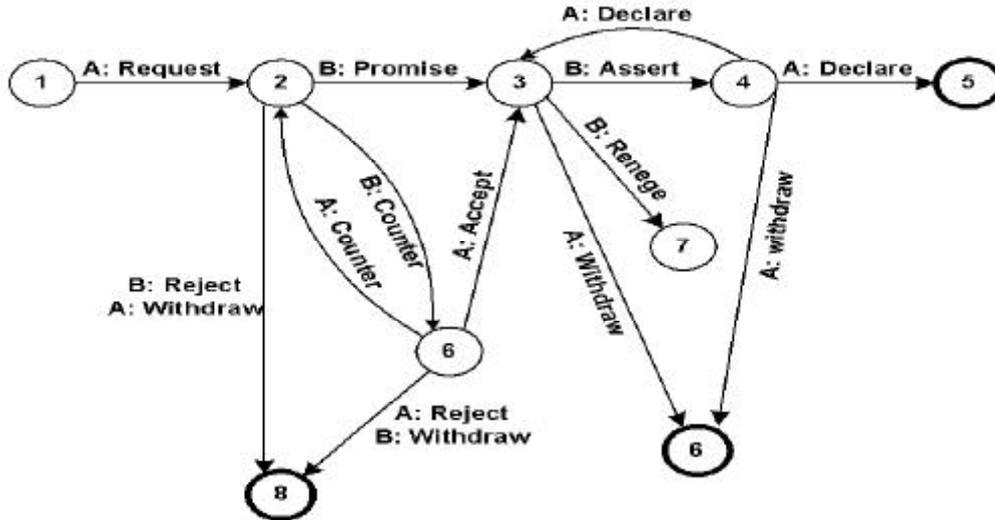


Fig. 3 Basic Conversation for Action

The Conversation for Action is a generic schema where successive speech acts are related to each other forming a network of speech acts like the one in [1]. Each circle represents a possible state of the conversation and arrows represent transitions accomplished by speech acts. With the request from initial speaker (A) to hearer (B), a transition is made from state 1 to state 2. In the above state transition diagram, there is a finite number of transitions that the conversation can take from a given state.

In the path showing successful completion of a conversation, B assert to A that the conditions of satisfactions have been met (state 4) and if A declares she is satisfied the conversation terminates successfully at the termination state 5. Note that there are also possible conversation failure termination states, for instance when a withdrawal of request from A leads to termination state 8 in the diagram.

The usage of pragmatic actions to the basic conversation for actions introduced above would be an interesting application to explore. The [5] shows the corresponding pragmatic actions for basic conversation for action.

According to the analysis adopted, the transition from state 2 to 6 is a proposal from Supplier to Customer for a change to an Economic Commitment. The transition from state 6 to 2 is a proposal from Customer to Supplier for a change to a counter-offered Economic Commitment.

Also the transition from state 4 to 9 has been mapped to two pragmatic actions where customer withdraws the established Economic Commitment and rejects the proposal Economic Event.

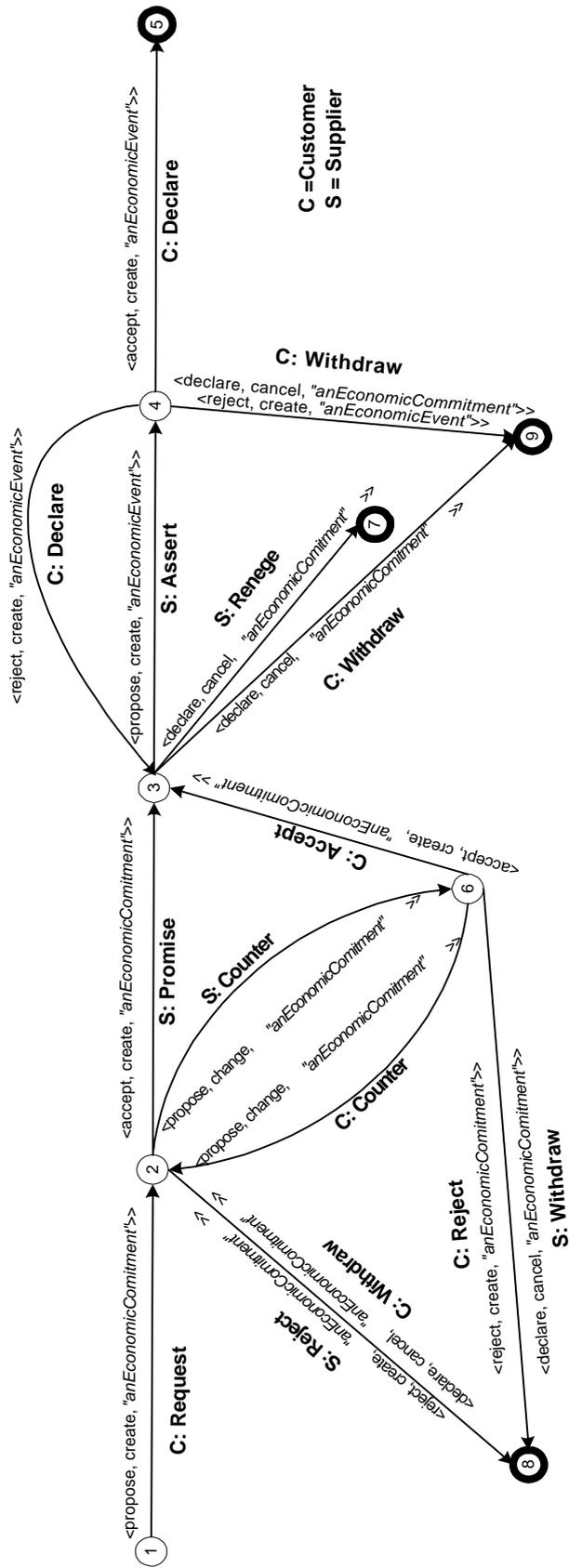


Fig. 4 Basic Conversation for Action (with Pragmatic Actions)

5.2 BUSINESS TRANSACTION AND COLLABORATION PATTERN ANALYSIS

UN/CEFACT has defined a number of business transaction patterns as part of UMM with the intention of providing an established semantics of frequently occurring business interactions. Below, we list and define a number of these patterns and show how they can be understood based on the framework introduced in the previous sections. Design patterns are defined as “descriptions of communicating objects and classes that are customised to solve a general design problem in a particular context” [8]. We will adopt this definition to the UMM transaction patterns and view a *transaction pattern* as a re-usable template of exactly one pair of a Requesting and Responding Business Activity customised to encode the intentions and effects of a business interaction in a certain context.

Definition: A transaction pattern (TP) is an activity diagram with two states designating the Requesting and Responding Business Activity. Every other state is an end state. All state transitions are represented and labelled by one or several pragmatic actions, each carried by the Requesting and Responding Business Activity, see Fig. 5- Fig. 6 and Table 1 below.

TP	Definition	Analysis
Commercial (Offer/ Accept)	“This design pattern is best used to model the ‘offer and acceptance’ business transaction process that results in a residual obligation between both parties to fulfil the terms of the contract. The pattern specifies an originating business activity sending a business document to a responding business activity that may return a business signal or business document as the last responding message.” [9].	Request: <Propose,Create,aContract>
		Response: <Accept,Create,aContract> or <Reject,Create,aContract>
Query/ Response	“The query/response design pattern specifies a query for information that a responding partner already has e.g. against fixed data set that resides in a database. The response comprises zero or more results each of which meets the constraining criterion in the query.” [9].	Request: <Query,None,anEffectType>
		Response: <Reply,None,anEffectType>
Request/ Response	“The request/response activity pattern shall be used for business contracts when an initiating partner requests information that a responding partner already has and when the request for business information requires a complex interdependent set of results.” [9].	Request: <Query,None,anEffectType>
		Response: <Reply,None,anEffectType> ¹
Request/ Confirm	“The request/confirm activity pattern shall be used for business contracts when an initiating partner requests confirmation about their status with respect to previously established contracts or with respect to a responding partner’s business rules.” [9].	Request: <Query,None,aCommitment/aContract>
		Response: <Reply,None,aCommitment/aContract>
Information Distribution	“This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt signal. The pattern is used to model an <i>informal</i> information exchange business transaction that therefore has no nonrepudiation requirements.” [9].	Request: <Assert,None,anEffectType>
		Response: Carries no pragmatic action
Notification	“This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt signal. The pattern is used to model a <i>formal</i> information exchange business transaction that therefore has non-repudiation requirements.” [9].	Request: <Declare,Create,aCommitment/aContract>
		Response: Carries no pragmatic action ² .
Fulfilment	The fulfilment pattern specifies the completion of an Economic Event [Fig. 5].	Request: <Propose,Create,anEconomicEvent>
		Response: <Accept,Create,anEconomicEvent> or <Reject,Create,anEconomicEvent>
Contract Proposal	The Contract Proposal Transaction Pattern is a variation of the aforementioned Offer-Accept transaction pattern where the Partners does not have to make their assertions of intentions legally binding. [Fig.4]	Request: <Propose,None,aContract>
		Response: <Accept,None,aContract> or <Reject,None,aContract>
Bilateral Cancellation	The Bilateral Cancellation transaction pattern refer to the bilateral cancellation of an Economic Contract or to Commitment(s) within an Economic Contract. See left part of [Fig. 6].	Request: <Propose,Cancel,aContract/aCommitment>
		Response: <Accept,Cancel,aContract/aCommitment> or <Reject,Cancel,aContract/aCommitment>
Unilateral Cancellation	The Unilateral Cancellation transaction pattern refers to the unilateral cancellation of an Economic Contract or to Commitment(s) within an Economic Contract. See right part of [Fig. 6].	Request: <Declare,Cancel,aContract/aCommitment>
		Response: Carries no pragmatic action

Table 1 Analysis of transaction patterns in terms of pragmatic actions

The analysis suggests one way to interpret the definitions of the UMM transaction patterns, but it does not make any claims to be the final, “correct” interpretation of these definitions. This is not an achievable goal as the definitions are only formulated in natural language, sometimes quite vaguely. The value of the

¹ Note that the analysis fails to make a distinction between the query/response and the request/response patterns; the reason for this is that the difference between the patterns does not reside in different business effects but in different ways of computing the responses.

² The motivation for this analysis is that a notification results in a binding specification of business conditions for the initiating partner and, thus, in a (partial) agreement.

analysis is that it provides explicit interpretations that can be judged for their validity, and thereby can help in formulating more precise and unambiguous definitions of the patterns.

Another use of the analysis is to suggest additional patterns than those already present in UMM. The Fulfilment, ContractProposal,, Bilateral and Unilateral Cancellations (from [Table 1]) are obvious candidates for business transaction patterns.

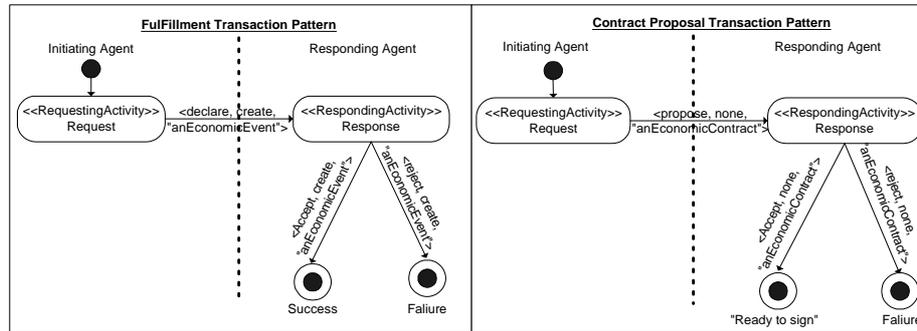


Fig. 5 Fulfilment and Contract Proposal Transaction Patterns

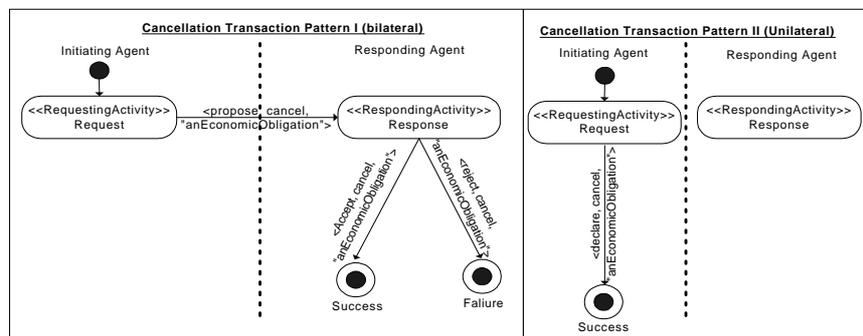


Fig. 6 Bilateral and Unilateral Cancellation Transaction Patterns

5.2 COLLABORATION PATTERNS

A *Business Collaboration Pattern* defines the orchestration of activities between partners by defining a set of BusinessTransactions patterns and/or more basic collaboration patterns plus the rules for transitioning from one transaction/collaboration to another [10]. The significance of a Business Collaboration Pattern is to serve as a predefined template in that it encodes business rules and business structure according to well-established good practices.

In UMM, some business logic in the collaboration patterns are, however, modelled on a programmatic level via instantiated parameters, e.g. as in the case of parameter 'isLegallyBinding', which takes two values, 'on' and 'off', distinguishing for instance between the real world concepts of proposal and offering of a contract. From a business modelling perspective this is difficult and cumbersome. Our analysis of the patterns make business intentions and effects explicit, i.e. present in the business model in their own right, in contrast to present in the encoding of the business model only.

Another problem with the UMM collaboration patterns is that their complexity increases dramatically as new patterns are assembled from basic patterns, making the resulting activity diagrams hard to understand. To overcome this difficulty we use a layered approach where the transaction patterns constitute nodes in the activity diagram of the collaboration patterns. In this way the internal interactions between business partners within a transaction are modelled in a set of well-defined transaction patterns. In the collaboration pattern this complexity is hidden, and only the outcome of the transaction pattern taken into consideration.

Definition: A collaboration pattern is a state chart over Transaction and Collaboration pattern(s). A collaboration pattern has exactly two end states representing success or failure of the collaboration respectively.

5.3.1 Fulfilment Collaboration Pattern

Definition: The fulfilment collaboration pattern specifies relevant transaction patterns (see Fig. 7) and the rules for transitioning among these within the completion of an EconomicEvent. The pattern is assembled from the Fulfilment and Unilateral Cancellation transaction patterns defined in the previous section.

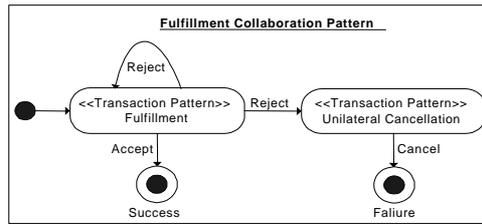


Fig. 7 Fulfilment Collaboration Pattern

5.3.2 Contract Proposal and Contract Offer Collaboration Patterns

Two basic collaboration patterns for business negotiation for contract formation are given in the Proposal and Offer collaboration patterns Fig. 8. The Proposal collaboration pattern models the non-legally binding negotiation phase in a contract formation whereas the Offer collaboration pattern expresses the formal creation phase of a contract, see Fig. 8. These patterns are assembled from the Contract proposal transaction pattern and Commercial transaction pattern (described in Section 5.1) respectively.

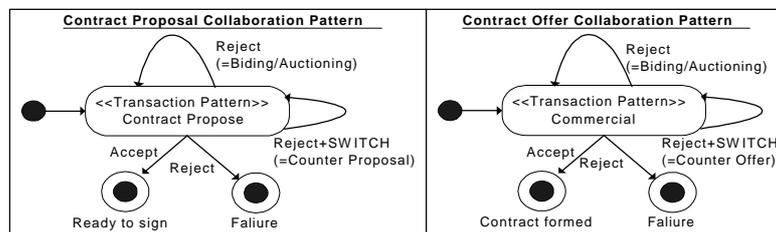


Fig. 8 Contract Proposal and Contract Offer Collaboration patterns

The two recursive paths when a contract offer/proposal has been rejected have a natural correspondence in the business concepts 'Counter Offer' and 'Bidding' (or 'Auctioning') respectively. 'Counter Offer' refers to the switch of roles between agents, i.e. when the responding agent has rejected the requesting agents offer, the former makes an offer of her own. 'Bidding' is modelled via the other transition from the decision activity, i.e. when the Responding Business Activity has turned down a contract offer, the Requesting Activity immediately initiates a new Transaction with a new (changed) offer for contract.

5.3.3 Composite Collaborations

More complex modelling and assembly of commitments, contracts and fulfilments are expressed in the example patterns found in Fig. 9.:

- Business Negotiation pattern
- Order-Fulfilment-Settlement pattern
- Long-term contract pattern with periodic releases
- Escalating commitments pattern
- Customer order direct delivery pattern

We will apply our framework for analysing the first collaboration pattern in the list above: the Business Negotiation pattern. This pattern is composed of, in turn, the previously defined transaction patterns: Query-Response transaction pattern, Contract Proposal and Commercial transaction patterns.

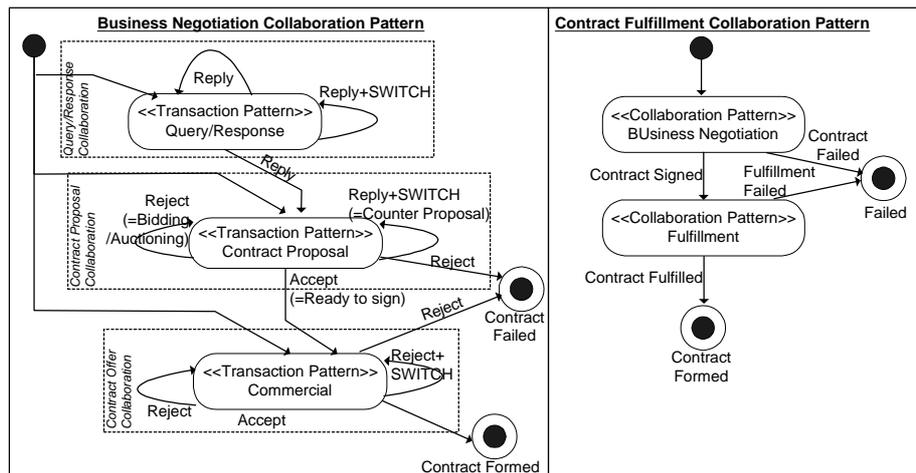


Fig. 9 Business Negotiation and Contract Fulfillment Collaboration Patterns

An example collaboration pattern composed of collaboration patterns only is also given in right part of the Fig. 9.

6 BENEFITS

- *Simplified Analysis and Design.* It will be easier for business users to participate in analysis and design if they are able to express themselves using concepts that have a business meaning (like propose, declare, commit, cancel) instead of using technical concepts like message structures and state machines. Furthermore, the specification of a pragmatic action is simple, as it can be viewed as filling in a template.
- *Technology Independence.* An approach based on pragmatic actions makes it possible to abstract business semantic conversations out of transaction and messaging protocols, so that pragmatic actions can be used with any capable technical collaboration protocol. Thus, pragmatic actions provide a clean interface to collaboration protocols.
- *Facilitating Teaching and Marketing.* Teaching and marketing a framework is greatly facilitated if you can explain and motivate it by telling a story. REA is eminently teachable, as you can easily tell a story that introduces its basic concepts. If pragmatic actions are included in UMM, it becomes much easier to tell a story, as you can move smoothly from physical message structures to communicative collaborations to economic effects.

7 OPEN ISSUES

6.1 WHEN DO THE REA OBJECTS COME INTO EXISTENCE?

In the text above, we have tacitly assumed that an REAObject is born as soon as someone proposes it. But the existence of an REAObject when it is only proposed may not be desirable in a shared environment. It could be better to introduce an “Instrument of Offer”, “Instrument of Acceptance”, etc. In a pragmatic action, these would be separate from the REAObject. A question here is whether we should add such instruments explicitly to the REAObjects. Some discussion on “Instrument of Offer” and “Instrument of Acceptance” can be found in UN/ECE Rec 31.

6.2 WHAT RELATIONSHIPS EXIST AMONG INTENTIONS AND PRAGMATIC ACTIONS?

Some relationships among Intentions are:

- a propose is associated to exactly one accept or reject
- a query is associated to exactly one reply
- a declare might be associated to an assert
- an assert might be associated to an assert

For actions:

- create must precede change
- create must precede cancel

These and additional relationships can assist a designer in verifying the correctness of a proposed collaboration.

7 RELATED WORK

One of the pioneers in speech act theory is John Searle, who developed a classification of speech act types in the 1960s. In recent years, he has extended his work on linguistic communication to the construction of a social and institutional reality.

According to Searle, a speech act is defined as an action changing the universe of discourse when a speaker utters it and a recipient grasps it. It may be oral as well as written, or even expressed via some other communication form such as sign language. Searle has developed speech act theory by introducing a taxonomy of five different kinds of speech acts: assertive, directive, commissive, expressive, and declarative, also called illocutionary points.

An *assertive* is a speech act the purpose of which is to convey information about some state of affairs of the world from one agent, the speaker, to another, the hearer. For example, the utterance “The father of speech act theory was Austin”. A *commissive* is a speech act, the purpose of which is to commit the speaker to carry out some action or to bring about some state of affairs. An example is the utterance “I will complete and submit the paper to REA’04”. A *directive* is a speech act, where the speaker requests the hearer to carry out some action or to bring about some state of affairs, e.g.. “You can complete and submit the paper to REA’04”. A *declarative* is a speech act, where the speaker brings about some state of affairs by the mere performance of the speech act. An example is the establishment of accepted papers, e.g. “Paper no 23 is accepted”. Finally, an *expressive* is a speech act, the purpose of which is to express the speaker’s attitude to some state of affairs, e.g. “I like the ideas presented in this paper”.

In addition to its illocutionary point, a speech act also has a propositional content. For instance, the speech acts “ I hereby pronounce you husband and wife” and “You are hereby divorced”, which are both declaratives, have different propositional contents. Furthermore, speech acts with different illocutionary points may have one and the same propositional content, which is the case with the examples for directive and commissive given above. A speech act is often viewed as consisting of two parts, its propositional content and its illocutionary force. The illocutionary force is the illocutionary point together with the manner (for example ordering, asking, begging) in which the speech act is performed and the context in which it occurs.

7 REFERENCES

1. Introductory survey on Searle: <http://artsci.wustl.edu/~philos/MindDict/searle.html>
2. Introductory tutorial on speech acts:
<http://www.rdillman.com/HFCL/TUTOR/Relation/relate2.html>
3. Recent book “The Construction of Social Reality”:http://www.amazon.com/exec/obidos/tg/detail/-/0684831791/ref=pd_bxgy_text_1/103-4670607-0647835?v=glance&s=books
4. Speech act theory has been applied to business communication by many researchers. One of the most comprehensive frameworks is FLBC, Formal Language for Business Communication. Link to FLBC site:<http://www.samoore.com/research/flbc/flbc.php>
5. An influential text on the relevance of communication and cognition for Computer Science is Winograd T. and Flores F., “Understanding Computers and Cognition: A New Foundation for Design”, Ablex, Norwood, N.J 1986 (It covers speech acts but is also much broader)
6. UN/ECE Recommendation No.31, ELECTRONIC COMMERCE AGREEMENT
http://www.unece.org/cefact/rec/rec31/rec31_2000_00tr257.pdf
7. Maria Bergholtz, Prasad Jayaweera, Paul Johannesson and Petia Wohead, "Process Models and Business Models - A Unified Approach", International WorkShop on Conceptual Modeling Approaches for e-Business, ER 2002 Tampere.
8. Gamma E., Helm R., Johnson R., and Vlissides J., “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison Wesley. 1995
9. “UN/CEFACT Modelling Methodology (UMM-N090 Revision 10), Valid on 20030328,
http://webster.disa.org/cefact-groups/tmg/doc_bpwg.html.
10. “eBTWG – Business Collaboration Patterns/Business Commitment Patterns Technical Specification (BCP²) – "Monitored Commitments"”, Valid on 20030401,
<http://www.collaborativedomain.com/standards/>