InterNeg

Satisfiability and Completeness of Protocols for Electronic Negotiations *

Gregory E. Kersten [1] and Hsiangchu Lai [2]

[1] J. Molson School of Business, Concordia University, Montreal, Canada
[2] National Sun Yat-sen University, Kaohsiung, Taiwan

Many of the existing e-negotiation support systems are build around one negotiation protocol. This effectively restricts their use to those problems and interactions that had been assumed a priori by the systems' designers. Field and experimental studies show that the way the negotiation process is structured depends on the negotiators' characteristics, the problem and the context in which an agreement is sought. It also has been recognized in literature that the way a problem is represented and the solution process implemented both strongly influence the results at which individual decision-makers and negotiators arrive. This article presents elements of a theory for the design of negotiation protocols. The proposed protocol formalism allows for the construction of models from which users can select a protocol instance that suits them and/or is appropriate for the problem at-hand. Furthermore, this formalism allows for the construction of protocols that can be modified during the user-system interactions. The paper also presents two key requirements for negotiation protocols embedded in support systems, namely their satisfiability and completeness.

Keywords: group decision and negotiation, electronic negotiations, electronic negotiation systems, negotiation support systems, protocols, rule-based formalism

* This work was supported with grants from the Initiative for New Economy of the Social Sciences and Humanities Research Council Canada, and the Natural Science and Engineering Research Council Canada.

Introduction

Internet and new software development technologies created new opportunities for the design and deployment of systems capable of supporting negotiators. Internet-based systems differ from other information systems in several key aspects including almost ubiquitous connectivity that allows a large number of people to access systems from almost any place. Internet-based systems also allow for tight integration of intra-enterprise business processes (e.g., supply chain management). Their user interface is provided by a client application, e.g. a web browser, which is easy to understand and common to many systems.

In terms of the implemented solutions and employed technologies Internet-based systems are unlike earlier systems deployed on local- and even wide-area networks. In the domain of e-negotiations some systems facilitate negotiation of the documents' content and their joint preparation, e.g., contract negotiations (Schoop and Quix 2001), others use email, chat and streaming video software (Moore, Kurtzberg et al. 1999; Lempereur 2004). Such systems as NegotiAuction (Teich, Wallenius et al. 2001) and GNP (Benyoucef, Keller et al. 2000) combine auction and negotiation elements. There are also systems that allow the negotiators to enter offers, which subsequently are sent to human experts who suggest agreements (e.g., cybersettle.com and electronicourthouse.com). A more detailed overview of e-negotiation systems (ENSs) can be found in Shim and Hsiao (1999) and Neumann et al. (2003).

In face-to-face (F2F) negotiations the parties may choose from among complementary activities and their control moves from activity to activity. The parties decide on the information they use, the strategies and tactics, decision rules, and so on. They also may undertake activities which they did not consider feasible at the beginning of the negotiation.

ENSs can play several roles in the negotiation process, including facilitating communication between the parties, providing decision support to the individual negotiator, aiding the parties in their search for agreements, and undertaking some or all activities autonomously (Kersten and Noronha 1999; Hämäläinen, Kettunen et al. 2001; Jennings, Faratin et al. 2001; Chen, Kersten et al. 2004). The flexibility that negotiators have in F2F encounters can be matched when communication systems (e.g., email and videoconferencing) are used. However, the more support there is from an ENS and the more actions the system undertakes autonomously, the more restrictions are imposed on the negotiators' own activities. The imposed restrictions are due to the *negotiation process model* and *negotiation protocol* implemented in the ENS.

In order to provide negotiators with meaningful support that is equip the ENS with mechanisms that help its users to achieve good agreements efficiently, it is important for the system to be based on a methodologically sound approach to negotiations (Lax and Sebenius 1986; Goldman and Rojot 2003), including the use of a negotiation process model (Gulliver 1979; Lewicki, Saunders et al. 1999). Negotiation process models comprise negotiation phases and assign a set of activities to each phase.

In F2F negotiation the selected process model provides the framework and helps the negotiators to orient themselves in their selection of strategies, tactics and specific activities. In e-negotiation the framework needs to be more precise; it needs to specify the activities undertaken on one hand by the negotiators and on the other by the ENS. The framework also need ensure that these activities effectively and efficiently contribute to the achievement of an agreement. It is the role of the negotiation protocol to provide this framework (Kersten and Lo 2003; Kim and Segev 2003). The protocol is a formal model, often represented by a set of rules, which govern software processing and communication tasks. The protocol imposes restrictions on activities through the specification of permissible inputs (Jennings, Faratin et al. 2001).

The comparison of F2F and electronic negotiation and the relationship between the key concepts is illustrated in Figure 1. The information that causes every e-negotiation activity to be undertaken is embedded in the protocol. The negotiation protocol selects the system's activities which either invoke the systems processing components or request input from the user. Information about the completion (successful or unsuccessful) of an activity is passed to the protocol which selects the next activity. In this way the protocol controls the ENS and also guides the users' activities.

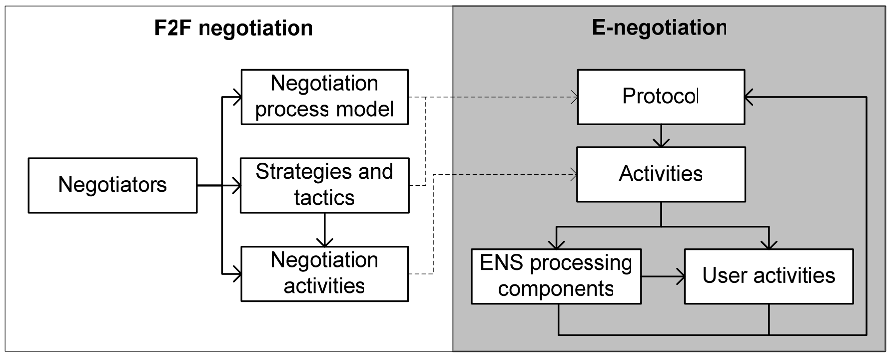

Figure 1. Face-to-face and e-negotiation concepts

Most of the ENSs implement only one negotiation protocol (Ströbel 2001). The two exceptions that we know are the SilkRoad platform (Ströbel 2003) and INSS (InterNeg 1997). SilkRoad has been designed to support various auction protocols; INSS was an early attempt to construct an ENS capable of supporting a few types of e-negotiations. One possible reason for the lack of systems in which multiple protocols are implemented and which can be easily configured to support new protocols has been lack of the common terminology and formal protocol models. The common terminology is now available in the Montreal taxonomy (Ströbel and Weinhardt 2003). The purpose of this paper is to present the theoretical foundations and discuss design and implementation issues of negotiation protocols. These protocols can be implemented in a software platform and then used for construction of various ENSs.

The importance of protocols is particularly visible in systems which actively participate in negotiations and which allow their users to select from among complementary activities. For example, the user may be able to submit one, two or more offers at the same time; she may submit a complete package or only values of selected issues; or she may make a conditional offer. Active systems may suggest one or more packages; propose that no offer be made; or suggest loosening soft constraints that restrict the values of one or more issues. In these situations the selection of a particular solution opens a new path of activities which have to be contiguous; every possible path selected by the user and/or system has to be purposeful and lead to the desired negotiation outcomes.

The remainder of the paper is organized as follows. Section 2 reviews protocol types and desired characteristics discussed in literature. Section 3 proposes a representation of negotiation processes based on negotiation phases, states and activities. The theoretical foundations of negotiation protocols and their properties are introduced in Section 4. In this section two equivalent representations are proposed and the protocol satisfiability and completeness conditions given. Section 5 presents ongoing and future work.

1. Protocol types and properties

In F2F negotiations the negotiators formulate, assess and revise objectives, aspiration and reservation levels, offers, arguments, threats and other concepts. Here they are called *negotiation constructs*; some of them are exchanged, others are private. In e-negotiations these constructs are jointly formulated by the negotiators, ENS and other systems (e.g., a DSS). We assume that, irrespectively of the source of construct formulation, the ENS can access every construct that is available to its users and thus it is able to differentiate among them.

The Montreal taxonomy (Ströbel and Weinhardt 2003) provides ENS designers with the terminology, description of the negotiation constructs, their roles in the process and the relationships between them. The importance of this taxonomy is also in that it allows describing the process and justifying the activities. This, in turn, makes it possible for the e-negotiation participants to select the specific e-negotiation they wish to conduct, know what the system is doing and why, what their tasks are and how they may contribute to the negotiation outcomes.

The negotiation process model provides an overall framework within which activities pertaining to the formulation of constructs and the constructs themselves can be positioned. The process of construct formulation and sequencing that takes place during e-negotiations is governed by the negotiation protocol.

Definition 1. *Negotiation protocol* is a model that guides processing and communication tasks of software and its users, and imposes—explicitly or implicitly—restrictions on their activities through the specification of permissible inputs.

Computer scientists working on the design of automated negotiations proposed using protocols to control software agents' negotiation activities (Cranor and Resnick 2000; Jennings, Faratin et al. 2001; Kraus 2001). Practical applications focus on the division and allocation of tasks and resources among computers and other systems (e.g., industrial printers, databases and robots). While the scope and degree of control of the interactions between users and the ENS need not be as complete and detailed as it is in automated negotiation, it is necessary to organize and schedule processing and communication tasks.

Negotiation protocols are not necessarily explicitly specified. In many early ENSs and also some recent systems the protocols are implicit; their users must follow the particular implementation of the e-negotiation process. To be able to move from one activity or task to another they have to provide information that the system requires. Neither the system nor its users can choose an activity but have to follow the pre-defined "hard-wired" sequence. For each of these systems, however, it is possible to re-construct and formally represent the protocol that the system and its users follow.

Several types of protocols are possible; a list of different protocols and their short explanations are given in Table 1.

Systems that support only completely structured exchange of information and disallow exchange of free-text messages are *closed*. Examples of these systems are auction systems in which the parties may submit only the issue values. Early ENS such as NEGO (Kersten 1985), RAINS (Hordijk 1991) and Web-HIPRE (Mustajoki and Hämäläinen 2000) are examples of systems in which implicit closed protocols were implemented. Closed protocols are also used in automated negotiations; with the negotiation software agents following the predefined set of decision rules.

Table 1. Types of e-negotiation protocols

Type	**Description**
Closed	All rules are defined a priori; no rule can be added or modified
Open	Rules may be constructed and added during the negotiation
Private	Guides the user's activities and defines her valid actions
Public	Defines the rules of interactions between the negotiators
Comprehensive	Can be used for different types of negotiations
Specialized	Applicable to one or a few negotiation types

A system that allows for exchange of free-text messages may follow a closed protocol. Its users, however, may introduce and agree upon new rules which they follow irrespectively of the rules that the system follows. For example, users of the Inspire system may ignore the system's components used for offer exchange and conduct the negotiation solely via the exchange of messages (Kersten and Noronha 1999). The system does not recognize these messages as offers and therefore it cannot use its analytical and graphical components. If the users achieve an agreement, Inspire does not recognize it and thus cannot assess its efficiency. In effect, while Inspire follows a closed protocol, users may follow an open one.

Thus, we need to distinguish between the *user perspective* and the *system perspective* on the negotiation protocols. From the user perspective most protocols of recently developed ENSs are open. With the exception of systems for on-line auction and bidding the use of open protocols will likely continue because of the need to provide the negotiators with systems that allow them to engage in unrestricted discussions. Because we are concerned here with the design of protocols for ENSs, the perspective we take is system-oriented.

From the system's perspective an ENS with open negotiation protocol requires a facility for the user or facilitator to construct and add new rules. Open protocols add complexity to the system's construction and use, but they may be required to account for the negotiators' learning and encountering problems that cannot be addressed using any existing element (e.g., rule) of the protocol.

It is useful to distinguish between *private* and *public* protocols. Private protocols are used to, for example, educate the negotiator, and help her select the strategy, evaluate counter-offers, make concessions and formulate arguments. Public protocols are used to set up the agenda, restrict the kinds of deals that the participants can make, impose a structure on messages, or specify the allowable sequence of offers and counter-offers. They may include the requirement that the negotiation be conducted in good faith or that an issue that both parties agreed upon be not renegotiated.

Another distinction of protocols is with respect to their comprehensiveness. *Comprehensive* protocols are those which can be used for several different types of e-negotiations. They allow using the ENS in which they are implemented for different negotiation processes and problems. There are few systems that implement comprehensive protocols. They deal with different types of auctions (Benyoucef, Alj et al. 2001; Ströbel 2003).

A *specialized* protocol is one that describes one or a few negotiation types and problems. Examples include SmartSettle, Inspire and WebNS (Thiessen, Loucks et al. 1998; Yuan, Rose et al. 1998).

E-negotiation protocols can be characterized by several desirable properties. Five such properties are presented in Table 2.

Table 2. Properties of e-negotiation protocols

Property	**Description**
Input consistency	All available information is considered for processing
Transparency	Users can observe and understand ENS behavior and actions
Explicability	Reasons for action selection are justifiable
Tractability	The purpose of every potential activity can be justified
Completeness	Interactions between the users and the ENS are sufficient to achieve the goal of the negotiation.

Input consistency means that no information available to the system is ignored. For example, if BATNA and the reservation levels are available, then they are both considered in assessing a counter-offer. If input is inconsistent the system has to resolve this inconsistency possibly with the help of the user.

Protocol *transparency* is required so that users may know what activity the system is undertaking and why. It does not necessarily mean that the role of every element of the protocol and every component of the system is understood by the user. Transparency also means that the user understands the sequence of the system activities and can position each activity in the negotiation process model. For example, a set of rules used to asses the counter-part as a hard negotiator is linked (through the opponent's assessment) with a set of rules used to make a concession.

The *explicability* property ensures that the activity selected by the protocol is traceable and the system can justify it.

The *tractability* property is more general than explicability; it refers to the protocol's capability to justify all activities that can potentially be undertaken.

The *completeness* property means that the activities undertaken by the user and the system under the guidance of the protocol lead to the negotiation goal, which is the achievement of the agreement or the realization that an agreement cannot be reached and the negotiation has to be terminated. Although this second goal may be considered a failure, it is no less important that the protocol allows for the process termination. This property assures that there are no "gaps" in the protocol so that for every negotiation situation there is an activity that can be undertaken so that the process continues. For every state in which the negotiators and the systems are, there is always at least one sequence of activities leading to the negotiation termination. The completeness property also assures that there is no activity that can be undertaken by the system that cannot be invoked by the protocol.

2. Phases, states and activities

E-negotiations and negotiations alike can be seen as a sequence of activities. The purpose of an activity is to formulate or reformulate the negotiation construct to which it pertains.

Some of the activities are necessary; they must be undertaken by the negotiators or the system for the negotiation to take place. To negotiate the negotiator has to consider (learn about) the problem, formulate and propose offers, assess counter-offers, and accept or reject a compromise proposal. Other activities may but do not have to be undertaken; the negotiator may write and send a message, attach an explanation to the offer or define aspiration levels.

Every activity takes place in a given *negotiation state* that belongs to one negotiation phase. In e-negotiations the activities are undertaken by the negotiators and by the systems they are using. Some activities may be combined, for example, the system may request the user to make an offer and write its justification, or the negotiator may, at the same time, view the counter-offer and read the negotiation problem. The relationship between the negotiation phases, states, activities and constructs is illustrated in Figure 2.

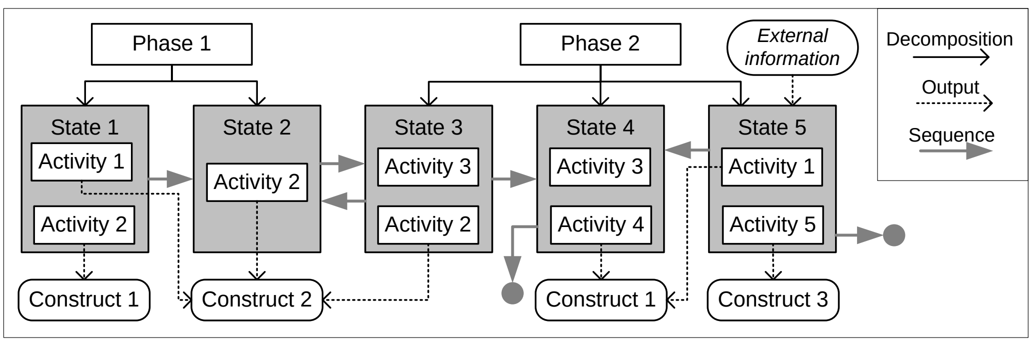

Figure 2. Phases, activities and constructs

A phase is decomposed into states; in every state one or more activities are undertaken. In general the activity's result is an output which specifies one or more values of a construct. Some activities, however, may not contribute to the construct specification, for example, the negotiator's confirmation of the offer submission and logging out from the negotiation.

One or more activities may be required to formulate the same construct. In Figure 2, activities 1, 2 and 4 are used to formulate construct 1.

At any point of the negotiation the user and the system are in one state; completion of all activities associated with this state moves the negotiation to another state. The completion of activities 1 and 2 moves the negotiation to state 2 and completion of activity 2 in state 2 moves the negotiation to state 3. As it can be seen, the same activities may appear in different states, for example, in two or more states the negotiator may write a message to his counterpart or read the negotiation problem.

Completion of some activities creates a situation when more than one move is possible. Upon the completion of activity 3, the move to either state 4 or state 2 of the preceding phase is possible (Figure 2). This allows the negotiator to cycle through a series of the same states; for example, the negotiator adds a negotiation issue, then formulates options for this issue, adds another issue, and so on.

The moves between phases and states (Figure 2) represent the process from the perspective of one negotiator. A similar representation may be constructed for other negotiators. The exchange of information between the negotiators invokes activities that the negotiator may undertake; external information activates state 5 in which activities 1 and 5 are undertaken. For example, an offer submitted by a counterpart invokes state in which the negotiator evaluates this offer (state 5). The negotiator, based on the counter-offer evaluation, moves to the state in which he constructs and submits an offer (state 4).

From the above description of the activities undertaken in various negotiation states it follows that states may be visited more than once. It also follows that the same activity may be undertaken in several states and the completion of an activity leads the negotiator and/or the system to the next state. The exception is the set of activities undertaken by the negotiator's counterparts.

The activities undertaken by the counterpart (e.g., activity "submit offer") are associated with the counterpart state. However, the information they produce activates the negotiator's state (e.g., in Figure 2, state 5 is activated by information provided by the counterpart).

It has been mentioned above that some negotiation activities have to be undertaken and others may but need not be undertaken. This distinction together with the distinction between the negotiator's and the counterparts' activities allow us to categorize negotiation states into the following three types:

1. *Mandatory states* are those which the user or the system has to visit in order to undertake another activity;
2. *Optional states* which activities may but do not have to be undertaken; and
3. *Intervening states* which are activated by information that is external to and not controlled by the user or the system.

The distinction between the three state-types is context-dependent. In the example presented in Figure 2, state 2 is mandatory for state 1, that is, the completion of activity 1 moves the negotiation to state 2. However, when the negotiation is in state 3, state 2 becomes optional; the negotiator (or the system) may either move to state 2 or to state 4. The negotiator may cycle between states 2 and 3, but to move to a new state at some point he has to select state 4. Thus, state 4 is mandatory for state 3.

3. Protocol representation

Several types of protocols were discussed above (Table 1). In all these types the negotiation states and activities occur. The movement from one activity to another keeps the negotiation process in motion. Based on the consideration of states and state transitions a formal protocol representation is given below. The initial focus is on closed and private protocols.

3.1 Preliminaries

In order to formulate protocol sequences and discuss protocol properties the following notation is used:

$\wp$ denotes a protocol;

$S = \{ s_1, \ldots, s_N, 0 \}$ denote the set of all states; state 0 is the null state;

$E = \{ e_i, i \in I^E \}$ – set of initial states;

$M = \{ m_i, i \in I^M \}$ – set of mandatory states; and
$O = \{ o_i, i \in I^O \}$ – set of optional states.

We also distinguish the subset of mandatory states $M^C = \{ m_i, i \in I^C \}$, ($M^C \subset M$). Every element of M^C concludes the negotiation; if the negotiator enters state m_i, ($i \in I^C$), the negotiation ends.

Based on the distinction between initial, optional and mandatory states we define a sequence of states.

Definition 2: A *sequence* comprises one initial "sequence entry" state, one mandatory "sequence exit" state, and zero, one or more optional states.

The relationship between the states in a sequence is defined as follows:

$$\sigma_i : (e_i \rightarrow \{m_i, O_i\}), \tag{1}$$

where:
σ_i is a sequence, $i \in I$, $|I|$ is the number of sequences;
e_i is the initial state in sequence σ_i, ($e_i \in E$);
m_i is the mandatory state in sequence σ_i, ($m_i \in M$); and
O_i is the set of optional states in sequence σ_i, ($O_i \subset O$).

When the user enters a sequence he may move from its initial state to any optional state and the mandatory state. He can repeat the activities associated with the optional, initial and mandatory states many times.

To move to a state that is not reachable from e_i of sequence σ_i, that is, to the state that is neither optional nor mandatory state in σ_i, the user has to move to mandatory state m_i. Only from this mandatory state the user can move to another sequence by entering its initial state.

Based on the above we can formulate conditions that allow us to distinguish states and specify the negotiation context.

Condition 1: The initial state e_i, ($e_i \in E$) uniquely defines sequence σ_i, ($i \in I$), that is: $|E| = |I|$.

When the negotiator and the system are in any given sequence they have to be able to move to at least one other sequence. The sequences that the negotiator can move to are *directly reachable*. The exception is the situation when the negotiator reaches a sequence in which the mandatory state concludes the negotiation. Thus we can formulate the following condition.

Condition 2: With the exception of states that complete the negotiation, every mandatory state m_i ($m_i \in M \setminus M^C$) of sequence σ_i, ($i \in I$) points to one or more initial states of sequences σ_j, ($i \neq j$; $i, j \in I$).

Condition 1 assures that every sequence has a different initial state; a sequence can be uniquely defined by its initial state. This does not mean that an initial state of a sequence cannot be an optional or mandatory state of another sequence. For example, the user enters the sequence in which the display of the negotiation problem is done in the initial state. When he moves to another sequence (e.g., exchange offers sequence), he may wish to read the problem again. But in this latter case the state associated with reading the problem activity is an optional state; the user may but need not to undertake it.

Condition 2 specifies that—with the exception of the states that complete the negotiation—every mandatory state of a sequence points to at least one other sequence. This allows for the selection of sequences from the mandatory state.

The possibility of selecting a sequence from any given sequence is required because it allows to cycle between sequences. It also allows to select from among alternative sequences associated with, for example, different models for preference elicitation and different communication modes.

Condition 2 introduces relationship between a mandatory state of one sequence and the initial states of at least one other sequence. Let E_i ,($E_i \subset E$), denote the set of the initial states of sequences other than sequence σ_i, which are directly reachable from σ_i, that is:

$$\forall\ \sigma_i,\ i \in I\ \ \exists\ E_i = \{e_j,\ j \in J_i\}. \tag{2}$$

From (2) follows that there are $|J_i|$ initial states that can be reached from the mandatory state of sequence σ_i. If there are no states that can be reached from a mandatory state, then this state concludes the negotiation. This allows the formulation of the following condition.

<u>Condition 3</u>: If $E_i = \varnothing$ in (2), then mandatory state m_i of sequence σ_i defined by (1) completes the negotiation, that is, $m_i \in M^C$. For every other sequence we have, according to Condition 2, $E_i \neq \varnothing$.

Formula (2) describes moves between sequences through linking the mandatory state of sequence σ_i with the initial states of sequences reachable from this state. In that (2) introduces the set of initial states that can be accessed from a sequence. This relationship between states in different sequences describes permissible moves between the sequences, that is between any given sequence and its directly reachable sequences. It is a new type of sequence, called here ρ-sequence, and it is defined by:

$$\forall\ i \in I : \rho_i = (m_i\ \rightarrow E_i), \tag{3}$$

where:

m_i ($m_i \in M$) is the mandatory state in sequence σ_i; and
E_i ($E_i \subset E$) is defined with (2).

Condition 3 provides the condition for mandatory states that complete the negotiation and those that allow the movement to other sequences. Taking into account (3), we see that this condition holds for ρ-sequence representation.

From Condition 3 and formulas (2)-(3) we obtain:

$$\forall\ m_i \in M^C\ \ (m_i\ \rightarrow \varnothing).$$

From Condition 1 and (2) it follows that (3) may be rewritten so it is clear that ρ-sequence describes moves between, rather than within, sequences. Because there are no possible moves from the states that conclude the negotiation we consider only mandatory states that point to at least one initial state:

$$\forall\ i \in I \setminus I^C : \rho_i = (\sigma_i\ \rightarrow \{\sigma_j,\ j \in J_i\ \}). \tag{4}$$

The relationship between σ-sequences is described with ρ-sequences. Every state that the negotiator and the system can visit is an element of at least one σ-sequences. We can thus represent protocol $\wp$ in terms of these two types of sequences:

$$\wp = \langle \{ \sigma_i, i \in I\}, \{\rho_j, j \in I \setminus I^C \}\rangle. \qquad (5)$$

The moves within and between sequences, and the three conditions restricting these moves are illustrated in Figure 3. The user first enters sequence B (after completing sequence A indicated with a circle). While in this sequence the user may move to two optional states unless he moves to the mandatory state in which he has two exit points to choose from. These exit points and sequences associated with them are examples of (3).

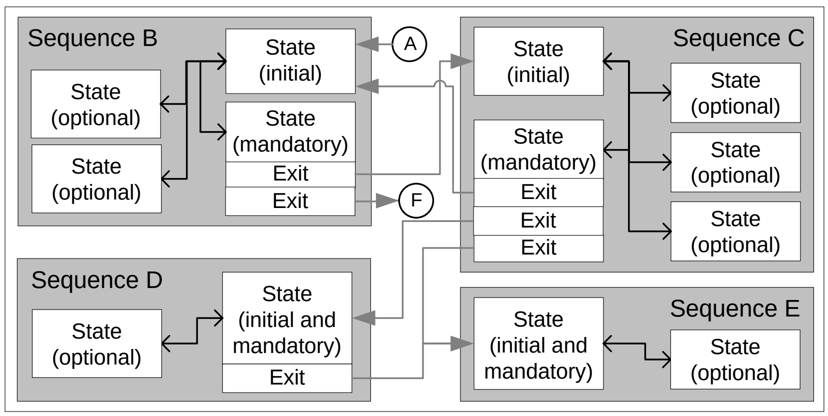

Figure 3. Moves between four sequences

One exit point in B moves the user to sequence C and another to F (not illustrated). In sequence C the user has three optional states that he can visit and the mandatory state with three exit points. One exit point moves the user back to sequence B. The second exit point moves the user to sequence D and the third to E.

Sequence D differs from sequences B and C because its initial and mandatory states are the same (we denote this state as “initial-mandatory”). This sequence has only one exit point that points to sequence E. This sequence has no exit points therefore E concludes the negotiation.

Optional states are included in all the four sequences illustrated in Figure 3. This is a typical characteristic of protocols for e-negotiations in which human negotiators participate; in every sequence the person should be able to access states that display the negotiation problem, their preferences, and the state used to terminate the negotiation. At the minimum, however, a sequence may comprise of only one state that is both initial and mandatory.

Without loss of generality we assume that the optional set O^i associated with initial state e_i is nonempty. In the minimum, the termination state (negotiation breakdown) is the optional state for every sequence given (1), that is:

$$\forall\, e_i \in E\ (i \in I) : O^i \neq \varnothing. \qquad (6)$$

Note that the termination state is generally different from the negotiation completion state. Negotiators may decide, at any point of the process, to breakdown the negotiation with their counterparts, seek other counterparts and begin new negotiation. A protocol should allow for the negotiation break down as an option. However, the negotiation completion states are not optional; they describe the situation in which the negotiators reached an agreement. There may be several negotiation completion states because the negotiators may complete the process immediately after reaching an agreement, try to improve it, and discuss issues pertaining to its implementation.

3.2 Sequences and rules

In Section 3.2 we said that negotiation protocols are defined in terms of "If … then …" rules but, in Section 4.1, it is proposed that negotiation protocols are represented by sequences. We now show that these two representations are equivalent.

We said that the negotiator and the system move from one state to another by performing activities associates with this state. Every state may be assigned a *truth* value through a valuation function, that is, if s is a state, then:

$$v : state \rightarrow \{true, false\}.$$

Sequence σ_i defined with (1) describes the situation when the entering initial state e_i allows to move to one or more of the optional states and to the mandatory state of σ_i. This can be denoted with the following "If … then …" rule:

$$v(e_i) \Rightarrow v(o_1^i) \vee v(o_2^i) \vee \ldots \vee v(o_k^i) \vee v(m_i), \tag{7}$$

where: $o_k^i \in O_i$; $m_i \in M$; $k = |O_i|$; $i \in I$.

From rule (7) it follows that when the truth value of $v(e_i)$ is *true*, then one or more consequent states may be selected and assigned value *true*.

When m_i is achieved (i.e., $v(m_i) = true$) then another rule of the protocol is invoked; this rule corresponds to ρ-sequence defined by (4). From (4) follows that from sequence σ_i it is possible to move to one of the directly reachable sequences. This move can be described with the rule representation of ρ-sequence:

$$v(\sigma_i) \Rightarrow v(\sigma_{j_1}) \vee v(\sigma_{j_2}) \vee \ldots \vee v(\sigma_{j_k}). \tag{8}$$

where: $i \in I \setminus I^C$; $\{j_1, j_2, \ldots, j_k\} = J_i$; $k = |J_i|$.

Rule (8) describes moves between σ-sequences; every sequence σ_j ($j \in J_i$) is directly reachable from σ_i. If the user and the system are in σ_i, (i.e., $v(\sigma_i) = true$), then they can move to one of the directly reachable sequences. The selection of sequence σ_j ($j \in J_i$, $i \in I$), means that $v(\sigma_j) = true$.

It should be noted that the antecedents in ρ-sequence defined with (8) are valuations of sequences σ_i, ($i \in I$) with the exception of those sequences that conclude the negotiation, that is, σ_i, ($i \in M^C$). This is because, according to (3), there are no reachable sequences from a negotiation concluding sequence.

From (7) and (8) it follows that the two types of sequences defined with (1) and (4) can be represented by rules. Protocol $\wp$ described with (5) can also be described with $|I|$ rules defined by (7) and $|I|$ - $|I^C|$ rules defined by (8).

Because the sequence notation is more compact than the rule notation the former is used in the following discussion of protocols, and their types and characteristics. The rule-based representation, however, is useful to prove protocols' properties.

The proposed here representation of protocols with two different types of rules (sequences) has an important implication for the specification of protocol characteristics and its assessment. There are two distinct levels of reasoning about protocols:

1. At the protocol-level only the relationships between σ-sequences are considered, the sequence composition is ignored. This means that at this level either (4) or (8) are considered. At this level protocol $\wp$ is seen as set of ρ-sequences $\{\rho_j, j \in J\}$.

2. At the sequence-level the composition of the σ-sequence comprising of states and their types (i.e., initial, optional and mandatory) is considered; states that are not elements of the sequence are ignored. This means that at this level either (1) or (7) are considered.

The two levels of reasoning about protocols are illustrated in Figure 3, in which both states and sequences are indicated. In order to illustrate reasoning at the protocol-level a protocol is schematically represented in Figure 4.

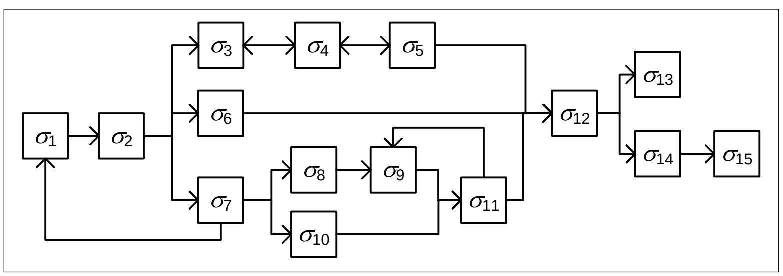

Figure 4. Example of a protocol with 15 sequences

The first four sequences for the protocol presented in Figure 4 are:

$$\rho_1 = (\sigma_1 \rightarrow \{\sigma_2\}),$$
$$\rho_2 = (\sigma_2 \rightarrow \{\sigma_3, \sigma_6, \sigma_7\}),$$
$$\rho_3 = (\sigma_3 \rightarrow \{\sigma_4\}),$$
$$\rho_4 = (\sigma_4 \rightarrow \{\sigma_3, \sigma_5\}).$$

At the protocol level, the σ-sequences are considered as black boxes. What is happening inside such a sequence is not relevant at this level. This includes the negotiation breakdown that can be invoked from any given sequence.

In the analysis at the protocol-level the main issue concerns the paths from any given sequence to some other sequence, in particular, from the first sequence of the protocol to each of the negotiation completion sequence. There are two such sequences (σ_{13} and σ_{15}) in Figure 4 and each of them completes the negotiation.

There are several cycles between the sequences, for example, from σ_7 it is possible to return to σ_1, and σ_3 from to σ_4. The possibility of cycles between σ-sequences introduces the problem of protocol completion, because, strictly speaking, the negotiation can take infinite time. When people participate in the negotiation it is obvious that there is a limit on the number of repeated sequences. No person can possibly engage in an infinite number of exchanges and other activities. Software agents due to limited intelligence and/or lack of a stopping rule could possibly undertake one or more activities an infinite number of times. This however is the agent design issue rather that of the protocol design.

Without loss of generality we assume that every cycle in a protocol can be executed a finite number of times so that the negotiator and system may achieve a completion state in a finite time.

3.3 Sequence modification

The mandatory state in a sequence may be a null state, that is:

$$e_i \rightarrow \{0, O_i\} \text{ with } e_i \in E, O_i \subset O. \qquad (9)$$

From (9) it follows that the user who entered state s_i cannot move to any other sequence and can undertake only activities associated with the optional states O_i and e_i. He will be able to move out of this sequence only when the null mandatory state is replaced with a state that points to other sequences as given by Condition 2. This replacement is due to external information received by the system.

An example of (9) is the offer exchange sequence which one negotiator entered but the other negotiator is still learning about the problem and therefore cannot read offers and make counteroffers.

After the negotiator who is in the offer exchange sequence submitted an offer, he has to wait for his counterpart to either accept this offer or propose a counteroffer. In the meantime, this negotiator may undertake some activities associated with optional states (e.g., review the case or analyze his preference structure) or submit another offer, but he cannot move to the reaching agreement sequence. The receipt of a counter-offer is associated with an intervening state which is discussed below.

If the user enters a sequence with null mandatory state, he has to wait for an intervening state to occur. The intervening state occurs when an activity is undertaken by an external entity and information about this activity is passed to the system. The external entity may be the negotiator's counterpart or an external system (e.g. a negotiation software agent).

When external information is received, the intervening state is immediately activated and the user is passed to this state. The intervening state then becomes:

1. An optional state in the sequence in which the user is at the moment and the sequences that follow this sequence; or
2. An optional state in the sequence in which the user is at the moment, the sequences that follow this sequence, and the initial-mandatory state in at least one subsequent sequence; or
3. An initial state in the sequence which immediately follows the sequence in which the user is when the external information has been received.

To illustrate the functions of the intervening states consider a negotiator who begins his negations some time later than his partner.

When this negotiator reads the problem for the first time, his counterpart sends a message. This message is immediately displayed to the user and the state in which the message is displayed becomes an optional state in this and subsequent sequences. The fact that the message display state becomes optional in subsequent sequences allows the user to access this message and possibly other messages that the counterpart sends. This example illustrates the first function of the intervening state; the external information (message) causes that the system inserts optional states to the current and subsequent sequences.

To illustrate the second function, consider the situation when the negotiator reads the problem for the first time and at this moment his counterpart makes an offer. The protocol implements a prescriptive negotiation approach which includes the specification of the negotiator's preferences. To avoid the possibility of the negotiator being influenced by the counterpart's offer, the system does not display the offer. Instead it informs the negotiator that an offer has been made and provides the reasons for not displaying the offer. This information is included in the optional state that is inserted in the current sequence.[1]

The optional state is inserted into every sequence the user may move to from the current sequence until he reaches the sequence in which offers are displayed. Sequences associated with preference elicitation and utility constructions are examples of the sequences in which the optional state (with information about the counterpart's offer) is inserted.

When the user moves to a sequence in which offers are displayed the first state the user is directed to the state in which the offers earlier received are displayed (the counterpart could send more than one offer). This is an example of the intervening state becoming an initial state in the offer exchange sequence. Because the user has an option to accept the offer this state is both initial and mandatory (like sequence D in Figure 3).

If the counterpart did not make an offer prior to the negotiator's entrance onto this sequence, the initial state in the offer exchange sequence would be different because the system did not receive an offer. For example, the initial state can be the state in which the user makes an offer and/or sends a message. Thus, the intervening state may replace the existing initial state in a sequence.

The negotiator who is in the offer exchange sequence cannot move out of this sequence unless: (1) he accepts his counterpart offer or (2) his counterpart accepts the offer that he made. The state in which the counterpart's offer is displayed is the mandatory state with an exit pointing to the agreement state which is the initial state of the next sequence. If the counterpart accepts an offer then the user is moved from the offer exchange sequence to the next sequence in which the agreement state is the initial state. This is an example of the third function of the intervening states.

The situation in which the negotiator receives an offer about which he is informed but which is not displayed is illustrated in Figure 5. Sequence A includes activities involving the negotiator's learning about the problem. The preference elicitation and utility construction activities are undertaken in sequence B. Because the offer has been received when the negotiator was in sequence, the optional intervening states were added to sequences A and B. Sequence C is the offer exchange sequence and the intervening state became the initial/mandatory state in sequence C.

---

[1] This optional state may also provide the user with the possibility to write his counterpart a message indicating that after he completes the preliminary activities he will study the offer and react to it.

The situation in which the negotiator is in the offer exchange sequence is illustrated in Figure 5, with sequences *D* and *E*. The negotiator moves between the initial state e/m_6 and the optional states O_4, and he cannot leave sequence *D* unless either he or his counterparts accept an offer. If the offer acceptance is received while negotiator is in sequence *D*, then he is moved to state e_7 in sequence *E*. The fact that the system rather than the negotiator initiates the move from one state to another is indicated with a black (rather than grey) arrow.

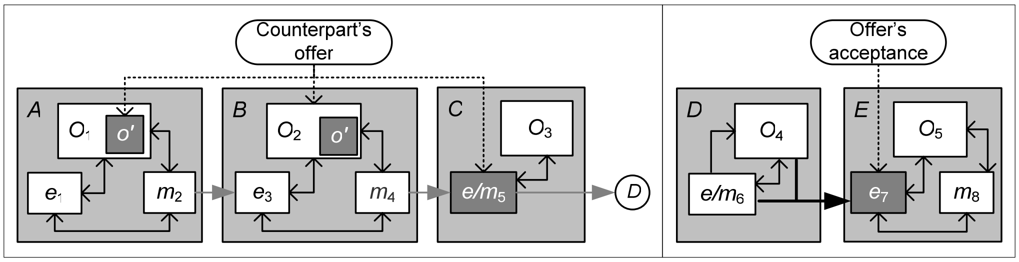

Figure 5. Functions of the intervening states

External information received by the system causes activation of an intervening state. What state is activated and to what sequences this state is added depends on the external information and on the sequence that the negotiator is visiting when this information is received. Therefore, we need to distinguish between different types of external information. Recall that the external information causes that the initial, optional and mandatory states are inserted in sequences.

Let *R* denote the set of information types and *P* the set of intervening states. With every element of *R* we associate three intervening states:

$$\forall\, r_l \in R : \{o_l, m_l, e_l\} \subset P, \tag{10}$$

where at least one of the states o_l, m_l and e_l cannot be null state.[2]

With every element of *R* we also associate the following three index sets:

$$\forall\, r_l \in R : I_l^O, I_l^M, I_l^E \tag{11}$$

where:

I_l^O is the index set of all sequences in which the optional state o_l is introduced;
I_l^M is the index set of all sequences in which the initial-mandatory state m_l replaces the existing initial and mandatory or initial-mandatory states (possibly null states), and
I_l^E is the index set of all sequences in which the initial state e_l replaces the existing initial state.

To represent the states' introduction and replacement caused by external information formulae (1) and (3) are used.

[2] It is possible that more than one optional state is associated with external information. This may be the case, for example, when different optional states are introduced to different sequences. The consideration of this situation is fairly straightforward and requires introducing sets of states rather than individual states.

Let denote the system's receipt of external information r_l as $v(r_l) = true$. Using (1), the occurrence of the external information causes the modification of one or more of the existing sequences, that is:

$$v(r_l) = true \Rightarrow \\ (\forall\, i \in I_l^O,\, o_l \neq \emptyset\ (e_i \rightarrow \{O_i, o_l, m_i\}) \wedge \forall\, i \in I_l^M,\, m_l \neq \emptyset\ (e_i \rightarrow \{O_i, m_l\}) \wedge \\ \forall\, i \in I_l^E,\, e_l \neq \emptyset\ (e_l \rightarrow \{O_i, m_i\})\ ). \tag{12}$$

Sequences which are modified in (12) can be represented as rules given in (7). Thus, (12) is a meta-rule; an occurrence of the external information causes the modification of one or more of the existing rules. The sequence modification involves addition of an optional state and/or the replacement of the existing mandatory and/or initial states.

External information and intervening states cause modification of sequences hence the protocol changes. These modifications occur in a sequential manner because we may consider information arriving sequentially. During the negotiation, the protocol is modified and states are added to the selected sequences. Thus we can distinguish the protocol before the parties begin the negotiation and then when they move through the states.

We can also distinguish between the protocol which is selected at the beginning of the negotiation, the protocol at any given time of the negotiation, and the *potential protocol*. The potential protocol is one that defines the negotiation providing that every type of external information that can affect the protocol is received. The potential protocol is thus one which is obtained from the initial protocol after modification defined by (12) occurred for every element r_l, ($r_l \in R$).

In the consideration of negotiation protocols the potential protocols are important because they define negotiations which can take place under the condition of the parties' willingness to exchange information.

3.4 Closed protocols

Closed protocols are those in which all rules are defined prior to the negotiation and no rule can be added or modified during the negotiation. This property means that the protocol's rules are defined a priori and no rules can be added or removed during the negotiation. In contrast, in open protocols new rules may be introduced or the existing ones—removed.

Open protocols require that the negotiators or other entities (e.g., the e-negotiation manager or the ENS itself) construct rules during the e-negotiation. It is advisable that these rules at least partially fulfill certain conditions which closed protocols meet. For example, an advisable property of a closed protocol is that the negotiation can be completed. In an open protocol this condition may be weakened to the condition of the negotiation progress in the sense that the addition of a rule allows the parties to move to another state that is required to complete the negotiation. Some requirements for closed protocols may thus be extended to open protocols. Using the notation introduced in Section 4.3 we formally define closed protocol.

<u>Definition 3:</u> Closed negotiation protocol is the 5-tuple:

$$\wp_C = (E, O, M, R, P). \tag{13}$$

Protocol $\wp_C$ is closed because its two types of sequences defined with (6) and (7), and their possible modification defined with (12) are known a priori and do not depend on the process.

Conditions 1-3 are necessary conditions for protocol $\wp_C$ to be closed.

Condition 1 states that $\wp$ contains only unique sequences; each sequence is defined by its initial state. If this condition is not met and two or more sequences have the same initial states then, using formula (7), it is not possible to uniquely determine the sequence that the negotiator and the system can move to. This means that the selection of a sequence cannot be done within the protocol, either additional rules are required to select the sequence or the users make the selection themselves.

Conditions 2 and 3 ensure that, with the exception of the sequences that complete the negotiation, it is possible to move from any given sequence to one or more sequences. If this condition is not met then the move from one to another sequence has to be determined using rules that are not part of protocol $\wp_C$.

The requirement for close protocols is that every negotiation completion sequence is reachable. The achievement of a completion sequence is equivalent to the conclusion of the negotiation through reaching an agreement.

In the assessment of closed protocols' properties the post-negotiation activities phase is not considered as it does not pertain to the process. Thus, we can identify the following three distinct negotiation conclusion states:
1. *Breakdown* state: no agreement is reached;
2. *Conclusion 1* state: agreement is reached and the activities from "Concluding the negotiation" phase are not undertaken; and
3. *Conclusion 2* state: activities are undertaken for the purpose of the revision, augmentation or further specification of the agreement reached in Conclusion 1.

The negotiation breakdown state may occur at any point of the negotiation; this state is optional and the negotiator may enter it if he does not wish to continue the particular negotiation instance. Conclusions 1 and 2 are similar in that both follow the agreement state. When Conclusion 1 is reached and the parties do not continue their negotiation, then the process ends with success (agreement reached). If however the process continues with the negotiators expect to reach Conclusion 2, the process may end with failure. Depending on the protocol, the breakdown state may be allowable in the "Concluding the negotiation" phase making it possible for the parties to terminate without the final agreement and thus not reaching Conclusion 2 state.

The key requirement for close protocols is that they assure the parties to be able reaching a negotiation completion state. Protocols that fulfill this requirement have the *completeness* property, which states that the interactions between the negotiators and the system are sufficient for the goal of the negotiation to be achieved, which is the expected negotiation conclusion. The goal is achieved if one of the conclusion states is reached.

Thus a complete protocol must ensure that there is a path comprising sequences that leads to the negotiation completion. This is the satisfability property and it has been used in logic regarding the assessment of well formed formulas (wffs): A well formed formula (wff) is *satisfiable* if it is not inconsistent, that is, if there are interpretations of the formula for which the formula is true.

Since protocols can be represented with rules which are wffs, we use formula (8) to provide necessary and sufficient conditions for a satisfiable $\wp_C$.

For every sequence σ_i ($i \in I$), and sequences σ_j ($j \in J_i$), which are directly reachable from σ_i, formula (8):

$$v(\sigma_i) \Rightarrow v(\sigma_{j_1}) \vee v(\sigma_{j_2}) \vee \ldots \vee v(\sigma_{j_k}).$$

can be rewritten as a set of rules:

$$R_i = \{\, v(\sigma_i) \Rightarrow v(\sigma_{j_1}),\ v(\sigma_i) \Rightarrow v(\sigma_{j_2}),\ \ldots,\ v(\sigma_i) \Rightarrow v(\sigma_{j_k})\}\}, \tag{14}$$

where: $i \in I \setminus I^C$; $\{j_1, j_2, \ldots, j_k\} = J_i$; $|J_i| = k$.

Formula (14) describes ρ-sequence using rules just as formula (3) describes it using states. From (5) and (14) we obtain the set of rules for all ρ-sequences of the protocol:

$$\Re = \{R_i,\ i \in I \setminus I^C\}. \tag{15}$$

In every protocol the existence of one *entry sequence* has to be assumed. The entry sequence's initial state is the first state of the protocol; the initiation of a negotiation requires that the user and/or the system enter this state. It is natural to assume a single entry sequence otherwise the choice of the first sequence is undefined.

The entry sequence has the same structure as other sequences defined in (1). Other sequences may point out to the entry sequence thus allowing the negotiator to revisit it at some point of the negotiation (this case is illustrated in 4).

Let σ_1 be the entry sequence of $\wp_C$.

Theorem 1: Protocol $\wp$ is satisfiable if, given $v(\sigma_1) = true$, there is a negotiation completion sequence which is a logical consequence of the set $\Im$ of formulae defined by (14)-(15), that is, if:

$$\exists\ k \in I^C : (v(\sigma_k) \Leftarrow \Re). \tag{16}$$

Proof: From (14) it follows that a negotiation completion sequence is not an antecedent of any rule in $\Re$. Thus it has to be a consequence in at least one rule in $\Re$, say this rule is $v(\sigma_{k-1}) \Rightarrow v(\sigma_k)$.

If sequence σ_{k-1} is not a consequent in any rule in $\Re$, then from $v(\sigma_1)$ and $\Re$, $v(\sigma_k)$ cannot be deduced. Therefore, there has to be a rule in $\Re$ in which $v(\sigma_k)$ is a consequent, say this rule is $v(\sigma_{k-2}) \Rightarrow v(\sigma_{k-1})$. Continuing, we obtain that there has to be a rule such that $v(\sigma_1) \Rightarrow v(\sigma_2)$.

From $v(\sigma_1) = true$ we obtain that $v(\sigma_2) = true$ and, continuing, we obtain that $v(\sigma_{k-1}) = true$ and thus $v(\sigma_k)$ is = *true*, ($k \in I^C$). That is, we have the set of k rules in protocol $\wp$:

$$\Re^k = \{\, v(\sigma_1) \Rightarrow v(\sigma_2),\ \ldots,\ v(\sigma_{k-2}) \Rightarrow v(\sigma_{k-1}),\ v(\sigma_{k-1}) \Rightarrow v(\sigma_k)\}. \tag{17}$$

The two properties of $\Re^k$ are:
1. With the exception of the completion sequence, the sequence that is the consequent of one rule is an antecedent of another rule; and
2. With the exception of the entry sequence, every antecedent sequence is a consequent in another rule.

Thus, from the entry sequence σ_1 it is possible to reach a negotiation completion sequence. ◊

When designing a protocol, satisfiability is a required property. Otherwise the protocol does not lead to any conclusion. In many situations, however, protocol completeness, which is much stronger property, should be sought.

Satisfiability guarantees that a conclusion may be reached if the path defined by (17) is followed. It does not say anything, however, about the negotiations which, at some point deviated from this path. The negotiator may thus end in a state which does not conclude the process but he cannot move from this state to any other state. Note that this possibility is not related to the negotiation breakdown—the situation when one of the negotiators decides to terminate the process. Rather, the issue here is that a negotiator may select a sequence from which no other sequence can be reached. Protocol completeness is the property that makes such a situation impossible.

The set of rules $\Re^k$ defines a path from the entry sequence σ_1 to sequence σ_k ($k \in I^C$) that concludes the negotiation. There may be more than one path from σ_1 to σ_k (Figure 4).

Let $\Re^*$ be the set of all possible paths from σ_1 to every concluding sequence σ_k, $\Re^* = \{\Re^l, l \in L\}$

<u>Theorem 2</u>: Protocol $\wp$ is complete if every sequence lies on at least one path from the entry sequence to a concluding sequence, that is:

$$\forall\ i \in I \setminus I^C\ \exists\ \Re^l \in \Re^* : (v(\sigma_i) \Rightarrow v(\sigma_-)) \in \Re^l, \tag{18}$$

where: σ_- is some sequence that is directly reachable from σ_i.

<u>Proof</u>: If $v(\sigma_i)$ is an element of $\Re^l$, then we obtain from (17) that: (1) from entry sequence σ_1 it is possible to reach sequence σ_i; and (2) from this sequence σ_i it is possible to reach a concluding sequence σ_k ($k \in M^C$). If valuation of every sequence in the protocol is an element of $\Re^l$, then this sequence is on the path from the entry sequence to some concluding sequence. ◊

Protocol completeness is the desired property of the potential protocol rather than the protocol at any point of the negotiation process. If a protocol is complete at some point in time there is no guarantee that it will remain complete when external information modifies its sequences. If a potential protocol is complete then we know that it is up to the negotiators to conclude the negotiation.

4. Conclusions

In this paper, we lay out theoretical foundations for negotiation protocols based on a negotiation methodology derived from behavioral research. The foundations facilitate the design and implementation of negotiation protocols and allow for the construction of ENSs based on these protocols. The construction of negotiation protocols may be a highly complex task. The theory given in Section 4 will allow for the implementation of a software tool that supports protocol designers and automatically verifies if the particular protocol meets the conditions formulated in Section 4.

A negotiation protocol is a specification of a sequence of activities that can potentially be undertaken by the negotiators and/or the system they use. The protocol can be specified by the user who participates in a specific negotiation, the system manager or the negotiation facilitator, or by the researcher who wants to study the use of the system's facilities, its efficacy, the relationship between the users' characteristics, the system usability, etc. This possibility to tailor the system's components to the needs of its users (researchers or managers) is done through the linking of the protocol-defined activities with the system components that either perform these activities or request the users to perform them.

The proposed approach allows for the construction of different ENSs based on the components and page composers which are associated with the components. These different ENSs are implemented in a software platform; a specific ENS is instantiated through the specification of the desired protocol instance. We are currently designing and implementing the software platform Invite, which will serve as a run-time environment for multi-protocol ENSs on the one hand and as a host for software tools for protocol design and verification on the other (Kersten, S. Strecker et al. 2004).

References

Benyoucef, M., H. Alj, M. Vezeau and E. K. Keller (2001). "Combined Negotiations in E-Commerce: Concepts and Architecture." Electronic Commerce Research Journal **1**(3): 277-299.

Benyoucef, M., R. K. Keller, S. Lamouroux, J. Robert and V. Trussart (2000). Towards a Generic E-Negotiation Platform. Proceedings of the Sixth International Conference on Re-Technologies for Information Systems, Zurich.

Chen, E., G. E. Kersten and R. Vahidov (2004). "Agent-supported Negotiations on E-marketplace." International Journal of Electronic Business: (to appear).

Cranor, L. and P. Resnick (2000). "Protocols for Automated Negotiations with Buyer Anonymity and Seller Reputations." Netnomics **2**(1): 1-24.

Goldman, A. and J. Rojot (2003). Negotiation. Theory and Practice. The Hague, Kluwer.

Gulliver, P. H. (1979). Disputes and Negotiations: A Cross-Cultural Perspective. Orlando, FL, Academic Press.

Hämäläinen, R. P., E. Kettunen, M. Marttunen and H. Ehtamo (2001). "Evaluating a Framework for Multi-Stakeholder Decision Support in Water Resources Management." Group Decision and Negotiation **10**(4): 331-353.

Hordijk, L. (1991). "Use of the RAINS Model in Acid Rain Negotiation in Europe." Environmental Science Technology **25**(4): 596-603.

InterNeg (1997). INSS: InterNeg Support System. Ottawa. 2004.

Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge and C. Sierra (2001). "Automated Negotiations: Prospects, Methods and Challenges." Group Decision and Negotiation **10**(2): 199-215.

Kersten, G. E. (1985). "NEGO - Group Decision Support System." Information and Management **8**(5): 237-246.

Kersten, G. E. and G. Lo (2003). "Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation." International Journal of Internet and Enterprise Management **1**(3): 293-315.

Kersten, G. E. and S. Noronha (1999). "Negotiations via the World Wide Web: A Cross-cultural Study of Decision Making." Group Decision and Negotiations **8**: 251-279.

Kersten, G. E. and S. J. Noronha (1999). "WWW-based Negotiation Support: Design, Implementation, and Use." Decision Support Systems **25**: 135-154.

Kersten, G. E., S. Strecker and K. P. Law (2004). A Software Platform for Multiprotocol E-Negotiations. InterNeg Working Papers. Ottawa, http://InterNeg.org.

Kim, J. and A. Segev (2003). A Framework for Dynamic eBusiness Negotiation Processes. IEEE Conference on E-Commerce, CITM.

Kraus, S. (2001). Strategic Negotiation in Multiagent Environments. Cambridge, MIT Press.

Lax, D. A. and J. Sebenius (1986). The Manager as Negotiator. New York, The Free Press.

Lempereur, A. (2004). "Updating Negotiation Teaching Through The Use of Multimedia Tolls." International Negotiations Journal **9**(1): (to appear).

Lewicki, R. J., D. M. Saunders and J. W. Minton (1999). Negotiation. Boston, MA, McGraw-Hill.

Moore, D., T. Kurtzberg, L. Thompson and M. Morris (1999). "Long and Short Routes to Success in Electrnically Mediated Negotiations: Group Affiliations and Good Vibrations." Organizational Behavior and Human Decision Processes **77**(1): 22-43.

Mustajoki, J. and R. P. Hämäläinen (2000). "Web-HIPRE: Global Decision Support by Value Tree and AHP Analysis." Information and Management **38**(3): 208-220.

Neumann, D., M. Benyoucef, S. Bassil and J. Vachon (2003). "Applying the Montreal taxonomy to State of the Art E-Negotiation Systems." Group Decision and Negotiation **12**(4): 287-310.

Schoop, M. and C. Quix (2001). "DOC.COM: A Framework for Effective Negotiation Support in Electronic Marketplaces." Computer Networks **37**(2): 153-170.

Shim, J. and N. Hsiao (1999). A Literature Review on Web-Based Negotiation Support System, Documentation for Web-based Negotiation Training System (WNTS). **2002**.

Ströbel, M. (2001). "Design of Roles and Protocols for Electronic Negotiations." Electronic Commerce Research Journal **1**(3): 335-353.

Ströbel, M. (2003). Engineering Electronic Negotiations. New York, Kluwer.

Ströbel, M. and C. Weinhardt (2003). "The Montreal Taxonomy for Electronic Negotiations." Group Decision and Negotiation **12**(2): 143-164.

Teich, J. E., H. Wallenius, J. Wallenius and A. Zaitsev (2001). "Designing Electronic Auctions: An Internet-Based Hybrid Procedure Combining Aspects of Negotiations and Auctions." Electronic Commerce Research **1**(1): 301-314.

Thiessen, E. M., D. P. Loucks and J. R. Stedinger (1998). "Computer-Assisted Negotiations of Water Resources Conflict." Group Decision and Negotiation **7**(2): 109-129.

Yuan, Y., J. B. Rose and N. Archer (1998). "A Web-Based Negotiation Support System." Electronic Markets **8**(3): http://www.mediamanagement.org/netacademy/publications.nsf/all_print_pk/1076.