Binding Telecooperation
- A Formal Model for Electronic Commerce -

Rüdiger Grimm, Peter Ochsenschläger

Institute for Secure Telecooperation

GMD Report 14.3.2000

Abstract

This paper formally defines the concept "electronic contract" and identifies its "goals", "obligations" and "binding phase". The definitions obtained here are used first for the specification of electronic contracts and secondly for the verification of local implementations of electronic cooperation contracts. The local representation of contracts and the communication between them, multiple and overlapping runs through a binding phase and the role of proofs are treated separately. The definitions are based on the theory of formal languages and automata. They are demonstrated by a simple example of a bilateral offer-order-deliver-pay cooperation.

1 Objective of the model

In general, human behavior cannot be specified completely. This also applies to goal-oriented cooperation in restricted application contexts such as in binding business transactions. Telecooperation technology aims at implementing the specifiable part of such cooperation contracts thus supporting the partners in collaborating.

We model business transactions as goal-oriented telecooperation activities performed by actors who assume specific roles. Roles are specified action patterns with defined goal states. The role-scripts contain non-deterministic branchpoints at which controlling and responsible persons take decisions considering semantic aspects within the framework of predefined action alternatives. For a detailed description of our telecooperation model, see [Gri94, 72 ff].

The exchange of goods and money is an example for a cooperative action pattern and its goals. The semantic purpose of the cooperation goals is not specified, it might be satisfaction through profit in our example. Like with games, a common syntactic goal is specified for every telecooperation as a regular end of cooperation. There is no semantic elaboration of the goal such as victory or defeat. The *goal* of cooperation is common to everybody, the *purposes* can differ or even conflict.

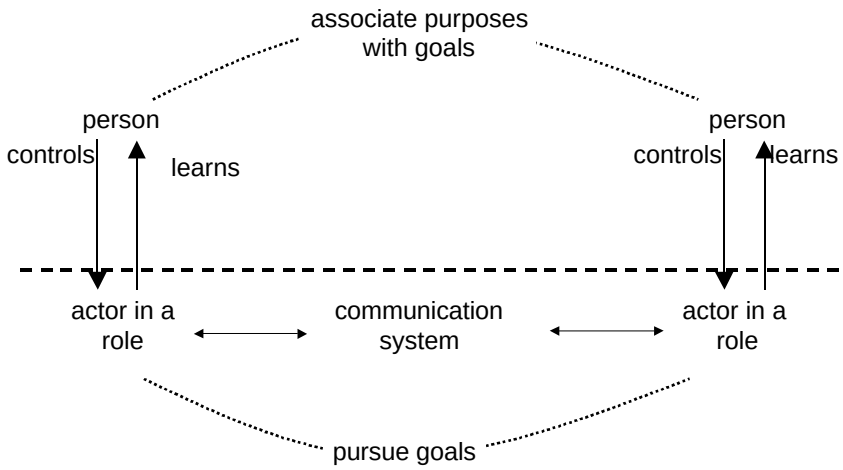

Fig. 1: Autonomous persons act as actors following specified role-scripts, i.e., in roles. The area below the dashed line is specified and also technically implemented if possible.

Every partner pursues his own specified individual goal: the buyer wants to receive the goods, the seller wants to receive the money. Our telecooperation contracts are basically designed such that either each of the participants reaches his (syntactic) goal or none of them. This coupling to success constitutes the cooperation principle of a common goal. It says nothing about the semantic purpose which somebody may associate with reaching his goal.

2 Formal languages, automata and language homomorphisms

The behavior L of a discrete system can be formally described by the set of its possible sequences of actions. Therefore L Φ^* holds where Φ is the set of all actions of the system and Φ^* the set of all finite sequences of elements of Φ, including the empty sequence denoted by ε. This terminology originates from the theory of formal languages, where Φ is called the alphabet, the elements of Φ are called letters, the elements of Φ^* are referred to as words and the subsets of Φ^* as formal languages. Words can be composed: if u and v are words, then uv is also a word. This operation is called the *concatenation;* especially $\varepsilon u = u\varepsilon = u$ holds. A word u is called a *prefix* of a word v if there is a word x so that v = ux. The set of all prefixes of a word u is denoted as pre(u); ε pre(u) holds for every word u. Formal languages which describe system behavior have the characteristic that pre(u) L also holds for every word u L. Such languages are called *prefix closed*. System behavior is thus described by prefix closed formal languages.

Formal languages can be represented by automata consisting of states (circles) and state transitions (directed edges). The edges are labeled by letters which represent actions. Actions are executed in conformance with the automaton by running the paths in the direction of the arrows. The associated letters form a word. An automaton accepts a word by processing the actions associated with the letters starting from a distinguished initial state and by reaching thus a final state. Every automaton defines a formal language in this way. If it is a prefix

closed language, all states are final states. Since the present paper is only to discuss prefix closed languages, final states will no longer be mentioned explicitly in the following.

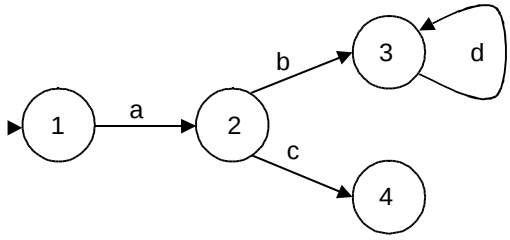

Fig. 2: Automaton which accepts the language of all words
{ε, a, ac, ab, abd, abdd, abddd, ...}.
Initial states are identified by arrowheads.

For the relationship between automata and formal languages, the set of the possible continuations of a word x L in language L is of great importance. It is formally expressed by the *left quotient* $x^{-1}(L)=\{y \quad \Phi^*|xy \quad L\}$ [Eil74]. In the language L={ε, a, ac, ab, abd, abdd, abddd, ...} (see Fig. 2), for example, the set of continuations of *ab* as well as of *abd* is equal to the set $\{d^n|n\geq 0\}$, therefore $(ab)^{-1}(L)=(abd)^{-1}(L)=\{d^n|n\geq 0\}$holds.The word *ac* can only be continued in L with the empty word: $(ac)^{-1}(L)=\{\varepsilon\}$. Such words are called *maximal* in L; max(L) denotes the set of all maximal words in a language L, i.e., $\max(L)=\{y \quad L|y^{-1}(L)=\{\varepsilon\}\}$.

In the automaton of Fig. 2, the states correspond uniquely to the left quotients of the accepted language L; $\{d^n|n\geq 0\}$, for example, corresponds to state 3. By means of this identification of left quotients and states, an accepting automaton can be constructed for every formal language [Eil74]; automata and formal languages therefore correspond to each other.

Mappings f: $\Phi^* \rightarrow \Phi'^*$ which are compatible with concatenation, for which therefore f(u)f(v)=f(uv) and f(ε)=ε hold, are called *language homomorphisms*. Language homomorphisms with the characteristic $f(\Phi) \quad \Phi' \quad \{\varepsilon\}$ are called *alphabetic* (since they map single letters onto single letters or onto the empty word). Abstractions of system behavior can be expressed by means of alphabetic language homomorphisms because they describe the hiding (f(a)=ε) and identification of actions (f(a)=f(b)).

If a system consists of several components which communicate with each other (distributed system), the alphabet of its actions is decomposed into disjoint subsets. In the case of two components, the behavior of the system is described by a prefix closed language $L \quad (\Phi \quad \Gamma)^*$ where $\Phi \cap \Gamma =$; the actions of the one component belong to Φ and those of the other component belong to Γ. Actions are uniquely assigned to the components executing them. By means of special homomorphisms (called projections) π_Φ: $(\Phi \quad \Gamma)^* \rightarrow \Phi^*$ and π_Γ: $(\Phi \quad \Gamma)^* \rightarrow \Gamma^*$ the local behavior $F \quad \Phi^*$ or $G \quad \Gamma^*$ of the individual system components can be extracted from the global system behavior: $F=\pi_\Phi(L)$ and $G=\pi_\Gamma(L)$. The projections π_Φ and π_Γ are defined by $\pi_\Phi(x)=x$ for $x \quad \Phi$ and $\pi_\Phi(x)=\varepsilon$ for $x \quad \Gamma$ as well as $\pi_\Gamma(x)=x$ for $x \quad \Gamma$ and $\pi_\Gamma(x)=\varepsilon$ for $x \quad \Phi$. [Och96] defines the language operation *cooperation product* which allows the global system behavior L to be represented by means of the local system behaviors F and G; this operation, of course, also reflects the communication system being used and the communication behavior of the components (cf. Section 9).

3 Example of an offer-order-deliver-pay cooperation and simplifying assumptions

In a simple offer-order-deliver-pay cooperation, a buyer and a seller exchange goods ("result") and money in accordance with specific regulations of a business contract. For this purpose, they use a communication system which guarantees that sending a message by one partner results in receiving the message by the other partner.

First, the business contract allows the exchange of general messages such as requests for offer, advertising material, greetings, inquiries and entertainment which are not binding for the two partners. Furthermore, the business contract prescribes the binding exchange of result and money in a sequence of predefined communication steps. In this example, we select the variant "first the result then the money". The associated messages are offer, order, result and money.

The buyer pursues the goal to receive the result (*r_result*), the seller pursues the goal to receive the money (*r_money*). To support a process of cooperation where either every partner reaches the goal or none, each of the partners takes on an obligation which brings the partner to the goal at the right moment. The seller is obliged to continue the sequence of actions *s_offer r_order* on his side by sending the result *s_result.* The buyer is obliged to continue the sequence of actions *s_order r_result* on his side by sending the money *s_money.*

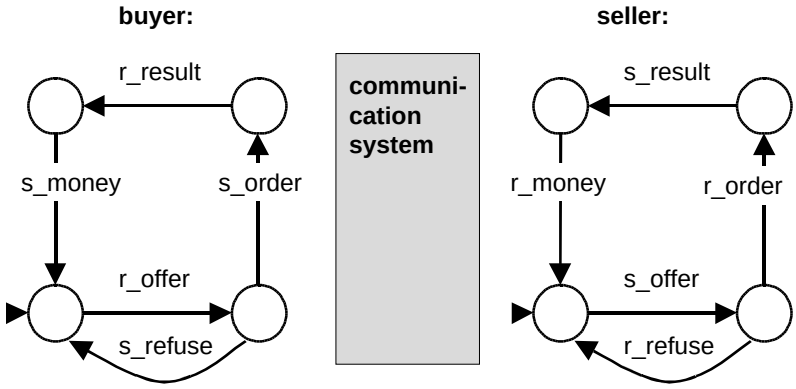

Fig. 3: Binding phase in the electronic contract of a simple offer-order-deliver-pay cooperation.

Notice the non-deterministic branching behind r_offer *on the buyer's side. The communication system can also be modeled as an automaton. It works such that every message which a partner sends is delivered to the other partner.*

In the interaction between obligations and goals, the ideal cooperation run is given by the global word *s_offer r_offer s_order r_order s_result r_result s_money r_money* where both partners reach their goals. On the other hand, other cooperation runs in which the buyer refuses or ignores offers are allowed, too. These runs are regular terminations in the sense of the terms of business. For the sake of simplicity, we will model only the explicit refusal of an

offer as a regular termination; other possible regular terminations (like ignoring an offer without replying) will not be considered.

Cooperation steps which are not binding for the two partners such as request for offer, advertising material, greetings etc. can also be admitted by the business contract, but will not be considered by the following presentation of our example for the sake of simplicity. That is, in our example, all cooperation steps are binding for at least one partner and therefore belong to the "binding phase" of the business contract.

We also assume here that the communication system is safe, i.e., that sending a message is bound to result in its receipt on the other side and that the receipt of a message is bound to be a result of sending it by the other side. In addition, we assume that all actions are non-repudiatably provable. There are technical mechanisms to implement these assumptions, for example, the digital signature (non-repudiation of origin) and acknowledgement procedures (non-repudiation of receipt) which are not discussed here any further. These simplifying assumptions allow the global sequences of *s_offer r_offer s_order r_order s_result r_result* ... to be reduced in a way to *offer order result* ... which are visible and non-repudiatably provable in the same way on both sides.

Section 10 is to refine this communication model by introducing a communication system explicitly. The "obligations" of the communication system are considered for the provability of actions on the other side.

4 Electronic contract

An *electronic contract EC* between two (ideal) cooperation partners F Φ^* and G Γ^* with $\Phi \cap \Gamma =$ is formally defined as a prefix closed language EC $(\Phi \Gamma)^*$ with the characteristics π_Φ(EC) F and π_Γ(EC) G.

An electronic contract is thus the definition of a set of global sequences of actions using actions from Φ and Γ whose projections to the one or other side constitute sequences of actions by F or G. That means: *F and G can behave in accordance with EC.* When considering F and G, one might often confine oneself to the contract behavior in EC and therefore one might postulate more strictly the equality: π_Φ(EC)=F and π_Γ(EC)=G. However, the somewhat more general formulation of the inclusion enables the description of more general partners F and G who can also behave outside a contract, for example, in order to process several contracts.

This global approach towards an electronic contract which considers the complete behavior of all sides is suitable for the specification of an ideal business process. In Section 9 we will pursue a constructive approach which first describes the local components and then integrates them into a coherent whole.

5 Binding cooperation

A cooperation is "binding" since it binds a promise (language) to the fulfillment (action) of this promise. A promise creates a state of open obligation. The state of obligation is kept open until the promise is fulfilled such that the associated obligation is dissolved.

In open communication environments such as in an open market of autonomous agents, obligations cannot be fulfilled simply by centrally controlled automatic mechanisms. It is therefore necessary to have binding cooperation described by *contracts.* The contract contains, for every partner, the *goals* he *intends* to achieve and the (conditional and unconditional) *obligations* he *is constrained* to fulfill. When a suitable structure of goals and obligations (of intent and constraint) is set up, the two partners, when acting in conformity with the contract, pres-

sure each other to the goal such that both partners achieve their goals. This "pressuring to goal" is typical of cooperation contracts.

In open environments, a principle of getting obligations fulfilled depends on an effective jurisdiction which requires the non-repudiatable provability of obligations. Cooperation protocols conforming with the contract are therefore characterized by a steady *balance between obligations and their proofs* [Gri94, 133 ff].

The basic idea of formalizing binding cooperation is the definition of binding phases V in contracts EC (Φ Γ)*. *Binding phases* V (Φ Γ)* *within EC* are prefix closed languages which are characterized by the fact that they consist only of *finitely many words* (*finiteness*) which, as far as they are not maximal in V, show *the same continuation behavior within V as within EC (closure)*. These two conditions of a binding phase mean that, once entered into a binding phase, one terminates within this phase.

Individual goals are sets of distinguished *subwords* within the binding phase. In a contract which supports the *cooperation principle* of common goals, the individual goals are structured such that every maximal word of the binding phase either reaches the goals of both partners or no goal (*success linkage*). The maximal words in V which achieve goals represent the cooperation runs which are successful (the buyer has got the goods, the seller the money). The maximal words in V which do not achieve the goals represent regular terminations, we call them regular aborts (the buyer keeps his money, the seller the goods, e.g. if no agreement on price could be achieved).

An individual *obligation* is a pair of a word followed by letters in the binding phase. The word represents conditions which contain promises. The following letters represent the fulfilling of the promises. By carrying out a word in an individual obligation, the conditional obligation contained in it is turned into an unconditional obligation and the remainder of the word (a letter) *must* now be carried out thus fulfilling the obligation.A contract supports the cooperation principle of common goalsif it fulfill the progress condition that its binding phases *are covered by obligations.*

Finiteness, closure, success linkage and *coverage condition* of a binding phase ensure the *pressuring to goal*: When a binding phase is entered, the obligations are processed in a finite number of steps. Due to the success linkage of the binding phase, either all the cooperation partners achieve their individual goals or no partner achieves any goal, even any subgoal.

Let EC (Φ Γ)* be an electronic contract. A prefix closed language V (Φ Γ)* is a *binding phase in EC* if the following holds:

(5.1) *Finiteness:* V is finite and $V \cap (\Phi \ \Gamma) \neq$

(5.2) *Closure:* x EC with x=yz and z V\(max(V) {ε})holds:
$x^{-1}(EC) \cap (\Phi \ \Gamma) = z^{-1}(V) \cap (\Phi \ \Gamma)$

In the "closure condition" (5.2), *y* is the (possibly empty) not binding segment and *z* the binding segment of x. Condition (5.2) says that regardless of past history *y* of a binding segment z, which together (x=yz EC) belong to EC, the following always holds: everything which can be done after that ($x^{-1}(EC)$) has to be done within V: $z^{-1}(V)$. That also applies recursively to every further step (letter) as far as no maximal word has been reached. Figuratively spoken: Words from EC which jut into V from now on remain and also end in V.

max(V) are all regulated terminationsof a binding phase. To recall: V only consists of a finite number of words by definition and therefore, there is a common finite upper bound to the length of *all* words in V: the maximum number of action steps to be done after entering V is always known in advance.

In the simple offer-order-deliver-pay cooperation, we only represented the words of the binding phase by omitting general messages such as request for offer *please_send_offer*, greetings, advertising material, entertainment which are possible within sales communication. Words which belong to the business contract, but which are within the binding phase only with their end segments, would, for example, be all words *please_send_offer s_offer r_offer ...* since *please_send_offer* is not binding for both partners.

max(V) here consists only of the two words *s_offer r_offer s_order r_order s_result r_result s_money r_money* and *s_offer r_offer s_refuse r_refuse*.

It is to be noted that, due to the cyclic specification of buyer and seller in Fig. 3, the binding phase can be passed as often as desired. That means:

EC is an *infinite* set of (finite) sequences of actions while V is a *finite* set, whose elements can also be executed repeatedly and successively as segments in the words of EC.

6 Goals

The success of a cooperation is associated with the achievement of goals by the cooperation partners. For this purpose, we define the *individual goals* Z_F for F or Z_G for G as subsets of $(\Phi \quad \Gamma)^* \setminus \{\varepsilon\}$. We can then formulate the "success linkage" such that the maximal paths of V either reach the goals of both partners completely or reach no goals, not even any subgoals of one of the partners. We require from a goal-oriented cooperation always that no partner pursues the empty word as an individual goal to enable every partner to head for an action. The formal definition is as follows:

"Individual goals" of F and G are subsets $Z_F, Z_G \quad (\Phi \quad \Gamma)^* \setminus \{\varepsilon\}$

If a goal contains more than one word, these words represent alternative goals ("or"). A goal will often contain only one single word. If a goal word contains more than one letter, these letters represent several actions to be taken in the given order ("and" with a fixed order). Subgoals are proper subwords of goals. If a partner has no goals of "his own" (e.g. obligation to furnish information), his goal can be identified with the partner's goal to avoid the forbidden empty word since we do not require an empty intersection from the partners' individual goals.

In the simple offer-order-deliver-pay cooperation, every cooperation partner has a one-letter word as goal: $Z_{buyer}=\{r\_result\}$ and $Z_{seller}=\{r\_money\}$.

The *achievement* of goals is defined by means of the projections $\pi_{\Phi Z} : (\Phi \quad \Gamma)^* \rightarrow \Phi Z^*$ and $\pi_{\Gamma Z} : (\Phi \quad \Gamma)^* \rightarrow \Gamma Z^*$ where $\Phi Z \quad \Phi \quad \Gamma$ and $\Gamma Z \quad \Phi \quad \Gamma$ are the sets of the letters of all words from Z_F resp. Z_G: A word $u \quad V$ achieves the goal Z_F or Z_G if $\pi_{\Phi Z}(u) \quad Z_F$ or $\pi_{\Gamma Z}(u) \quad Z_G$, i.e., if it contains a complete word (complete goal) of Z_F or Z_G as a subword.

An *individual goal path* of F is a maximal word of V which achieves Z_F. By analogy for G, an *individual goal path* of G is a maximal word of V which achieves Z_G. The sets V_{ZF} and V_{ZG} of the individual goal paths of F and G, respectively, are defined by

"Individual goal paths" $V_{ZF} := \pi_{\Phi Z}^{-1}(Z_F) \cap \max(V)$ and $V_{ZG} := \pi_{\Gamma Z}^{-1}(Z_G) \cap \max(V)$

A *common goal path* is distinguished by achieving *both* the goal Z_F *and* the goal Z_G (completely):

"Common goal paths" $V_Z := V_{ZF} \cap V_{ZG} \qquad (= \pi_{\Phi Z}^{-1}(Z_F) \cap \pi_{\Gamma Z}^{-1}(Z_G) \cap \max(V))$

Strictly complementary to that, a *regular abort* is distinguished by not containing a single letter of a goal, i.e. by not achieving any subgoal of F or G :

"Regular aborts" $V_A := \pi_{\phi Z}^{-1}(\varepsilon) \cap \pi_{\neg Z}^{-1}(\varepsilon) \cap \max(V)$

$V_A \cap V_Z =$ now holds since, if x V_Z, then $\pi_{\phi Z}(x)$ Z_F (and $\pi_{\neg Z}(x)$ Z_G) and since ε Z_F (and ε Z_G), $\pi_{\phi Z}(x) \neq \varepsilon$ (as well as $\pi_{\neg Z}(x) \neq \varepsilon$), hence x V_A.

Further, V_A V_Z max(V) holds, and, in general, this inclusion is proper. That is, in general, there are maximal paths with which one partner achieves his goal without the other partner achieving his goal either. Success linkage requires equality, i.e., that, also vice versa, a maximal path belongs either to V_A (in this case, it achieves no goal at all, not even a subgoal) or to V_Z (in this case, it achieves the goals of all partners completely):

(6.1) *Success linkage:* V_A $V_Z = \max(V)$

In the simple offer-order-deliver-pay cooperation, max(V) contains only the two words *s_offer r_offer s_order r_order s_result r_result s_money r_money* and *s_offer r_offer s_refuse r_refuse*. The first word achieves both the individual goal of the buyer {*r_result*} and the individual goal of the seller {*r_money*} and therefore, it belongs to $V_Z = V_{ZF} \cap V_{ZG}$. The second word achieves neither the one nor the other goal and therefore belongs to V_A. Success linkage therefore applies to this example.

The following holds for binding phases with linkage success:

(6.2) V_A $V_Z = \max(V)$ $V_Z = V_{ZF} = V_{ZG}$ (inversion does not hold in general)

(6.3) V_A $V_Z = \max(V)$ $\Leftrightarrow$ $V_A = \max(V) \setminus \pi_{\phi Z}^{-1}(Z_F) = \max(V) \setminus \pi_{\neg Z}^{-1}(Z_G)$

Proof of (6.2): Let V_A $V_Z = \max(V)$ and x V_{ZF}. Due to ε Z_F, $\pi_{\phi Z}(x) \neq \varepsilon$ holds, that is x V_A. Since, however, V_A $V_Z = \max(V)$, x V_Z must hold, and since, by definition, $V_Z = V_{ZF} \cap V_{ZG}$, then x V_{ZG}, hence V_{ZF} V_{ZG}. The inverse inclusion results by analogy.

Equivalence even applies to goals of length 1 in (6.2). For goals of greater length, however, which contain proper subgoals, the inversion of (6.2) does not hold in general since the equivalence of maximal paths which achieve individual goals completely, does not exclude that there are paths which, though achieving subgoals, do not achieve goals completely.

(6.3) is proved by means of simple set theory considerations.

A cooperation with success linkage now has these desired characteristics: First, maximal words of V can achieve goals. Secondly, there is no maximal word in which a partner achieves a subgoal without him and his partner achieving their goals completely, and if no goal is achieved, cooperation is aborted regularly.

7 Obligations

Though the success linkage defined in the preceding section ensures that each or none of the contracting parties achieve their goal after a sequence of actions from max(V), it is by no means excluded that a partner gets out of the cooperation prematurely (before max(V)) since his goal has been achieved, for example, though his partner's goals has not been achieved as yet. To avoid such a getting out, the partners are *obliged* to take specific follow-up actions.

An *obligation* of a partner is a conditional proposition: "if so-and-so actions have occured at this partner, he must continue with so-and-so actions". The first part is the prerequisite for a conditional obligation and represents the promise, the second part represents its fulfillment.

A contracting party can be obliged, of course, to take only such actions which are "within its responsibility", typically these are sending actions: After the receipt of a purchase order, a seller is obliged to deliver the goods. Receiving actions, on the other hand, are "in the responsibility" of the communication system: The communication system is responsible for delivering a submittedmessage or product to the receiver. Therefore, the obligations of the communication system have also be considered in addition to the obligations of the contracting parties.

For formalizing these concepts, we decompose the alphabets Φ and Γ of the two contracting parties F Φ* and G Γ* into two disjoint classes Φ=Φ Φ^0 and Γ=Γ Γ^0 each, where Φ and Γ are in the responsibility of F or G while Φ^0 and Γ^0 are both in the responsibility of the communication system. The -actions are also to be referred to as the *active* actions of the corresponding partner while we speak of the *passive* actions of the corresponding partner in the case of ^{0}-actions.

In addition, we introduce the terms Ω =Φ^0 Γ^0, Ω^0=Φ Γ and Ω= Ω Ω^0 to describe the communication system whose active actions are the passive actions of the contracting parties and whose passive actions are the active actions of the contracting parties. Ω ∩Ω^0= also holds for this alphabet. All the obligations of the communication system can then be formulated analogously to those of the contracting parties. The complete alphabet of the communication system consists of the letters of Φ and Γ. We observe the communication system so to speak from the viewpoint of its causes and effects for the partners F and G.

If an electronic contract EC (Φ Γ)* contains "internal" actions of the contracting parties, i.e., actions which do not affect the communication system, Ω^0 can be taken as a subset of Φ Γ instead of Ω^0=Φ Γ . Φ \Ω^0 or Γ \Ω^0 is then the set of internal actions of F or G. The behavior K Ω*=(Ω Ω^0)* of the communication system in EC is given by K=π$_Ω$(EC).

The set of the obligations of F is a set O$_F$ π$_Φ$(V)× (Φ) with the characteristics (7.1) and (7.2), where x π$_Φ$(V) denotes a promise and M Φ the alternative ("or"-ed) obligations to fulfill the promise ((Φ) denotes the power set of Φ). Obligations O$_G$ π$_Γ$(V)× (Γ) for G and O$_K$ π$_Ω$(V)× (Ω) for the communication system K are defined analogously. In formal terms:

The *set of the obligations* of F is a set O$_F$ π$_Φ$(V)× (Φ) with the characteristics

(7.1) (x,M) O$_F$ and y π$_Φ^{-1}$(x)∩V hold: x≠ε and y^{-1}(V)∩M=y^{-1}(V)∩Φ ≠ . That means: everything from Φ what may occur after a promise *x* is located completely in M ("=") (if necessary after intermediate steps on the other side) and there is also something to be done in M ("≠ ").

(7.2) (x,M) O$_F$ and m M y π$_Φ^{-1}$(x) with ym V.
That means: every fulfillment *m* depends on a promise *x* which in common belong to a valid word *ym* of the binding phase; with *y* possibly containing intermediate steps of the other side.

The condition (7.1) is essential for the definition of obligations; if there is a (x,M) π$_Φ$(V)× (Φ) which fulfills (7.1), then the M is uniquely defined by x, Φ and V due to (7.2).

We say: *F is under the obligation M after a sequence of actions y∈V* if there is a (x,M) O$_F$ with π$_Φ$(y)=x. Because of (7.1), there is then a m M with ym V; i.e., *F fulfills his obligations M with the sequence of actions ym∈V.*

Let Φ be the alphabet of the buyer and Γ the alphabet of the seller in the simple offer-order-deliver-pay cooperation. Buyer and seller have each an obligation. The following is true for the seller: *If* the seller makes an offer *and if* he receives the relevant order,

x_1 = *s_offer r_order* $\pi_\Gamma(V)$, he *then must* deliver the goods in question, M_1={*s_result*} (Γ), i.e.,

$$(x_1,M_1) = (s\_offer\ r\_order, \{s\_result\}) \quad O_{seller} \quad \pi_\Gamma(V) \times (\Gamma)$$

is the seller's single obligation.

The following is true for the buyer: *If* the buyer places an order *and if* he receives the relevant goods, x_2 = *r_offer s_order r_result* $\pi_\Phi(V)$, he *then must* pay them, M_2={*s_money*} (Φ), i.e.,

$$(x_2,M_2) = (r\_offer\ s\_order\ r\_result, \{s\_money\}) \quad O_{buyer} \quad \pi_\Phi(V) \times (\Phi)$$

is the buyer's single obligation.

If F is under the obligation M after a sequence of actions y V, the definition ensures that there will be a follow-up action Φ . It does not exclude, however, that there may also be follow-up actions from Φ, namely from Φ^0 which are not under M. This is the case if $y^{-1}(V) \cap \Phi^0 \neq$. Typically, these are receiving actions against which F cannot defend himself anyway.

The actual meaning of an obligation is that a contracting party or the communication system *must perform* a corresponding (active or possibly passive) follow-up action if he or it is under an obligation after a sequence of actions. Note that the definition is compatible with the fact that more than one participant, i.e., not only one partner or not only the communication system, can be under an obligation.

8 Covering the binding phase through obligations and pressuring to goal

With respect to success linkage we introduced obligations with the aim to ensure the continuation of specific sequences of actions which belong to V but not yet to max(V) so that each or none of the partners achieves his goal. We must make sure by obligations that all sequences of actions from $V \setminus ((\pi_{\Phi Z}^{-1}(\{\varepsilon\}) \cap \pi_{\Gamma Z}^{-1}(\{\varepsilon\}))$ max(V)) are continued:

The obligations O_F , O_G and O_K *cover the binding phase V completely* if

(8.1) after every sequence of actions y $V \setminus ((\pi_{\Phi Z}^{-1}(\{\varepsilon\}) \cap \pi_{\Gamma Z}^{-1}(\{\varepsilon\}))$ max(V)), F, G or K are under an obligation.

This definition actually covers only the *goals* completely. This is sufficient for securing the pressuring to goal defined below. Though *no* goal is contained in *s_offer r_offer s_order r_order*, it is a complete promise. The formulation in (8.1) "y V but not in $(\pi_{\Phi Z}^{-1}(\{\varepsilon\}) \cap \pi_{\Gamma Z}^{-1}(\{\varepsilon\})$" therefore holds only if a first subgoal has been achieved. In practice, however, there are obligations before that which are to force achievement of the very first goal as soon as a certain state of the cooperation has been reached.

Together with success linkage, this definition implies immediately the following proposition which we also call *"pressuring to goal":*

If the obligations O_F , O_G und O_K cover the binding phase V completely and if the success linkage of goals applies to the complete binding phase, then either each or none of the contracting partners has achieved his goal after every sequence of actions y V after which neither F nor G, nor K is under an obligation.

According to the definitions, this proposition can be derived directly by means of minor transformations [GO99]. Nevertheless, it is a fundamental proposition and all the preceding defi-

nitions have been developed to formulate it. The special thing about it is that the definitions of the binding phase, its complete coverage through obligations and the success linkage of its goals are characteristics which are sufficient for realizing the pressuring to goal. Furthermore, they also illustrate the deeper meaning of a contract-conforming cooperation.

From the viewpoint of the engineer, this proposition shows how to specify a contract-conforming cooperation protocol: when specifying a binding phase, we only need realize its complete coverage through obligations and success linkage in order to achieve the essential characteristic of a contract, its pressuring to goal.

This proposition is called "pressuring to goal" because a follow-up action is ensured through the obligations for every sequence of actions y V\((πΦZ^-1({ε})∩πΓZ^-1({ε})) max(V)) until an element from V_Z is reached.

For clarifying controversial cases, i.e., if cooperation does not make progress as planned, every cooperation partner as well as the communication system must be able to prove non-repudiatably, which actions have taken place with him or it. Therefore, *proofs* have to be collected; in formal terms, these are elements of $\pi_\Phi(V)$ for F, $\pi_\Gamma(V)$ for G and $\pi_\Omega(V)$ for K. Section 13 will discuss the handling of proofs in more detail.

9 Cooperation products and realizations of electronic contracts

In the above section, we have assumed that the specification EC of the electronic contract is available in global form. As long as it is a requirements specification for a distributed system as in the present case, this is quite usual. In the case of implementation specifications in open communication environments such as the Internet, however, only local specifications of the various components and the communication system are usually available. By using *cooperation products* as defined in [Och96], the global system behavior can be obtained from these in form of a prefix closed language. With the definition of an obligation-conforming realization (see below) we develop a tool for verifying local implementations.

On the basis of the local behavior patterns F Φ* and G Γ*, π_Φ(EC) F and π_Γ(EC) G hold necessarily for the global system behavior EC (Φ Γ)*. EC depends, however, not only on F and G but also on the kind of communication. As an example, let us consider a commonbuffer of capacity 1 which can accept and deliver the messages *offer, order, refuse, result* and *money*; its behavior is described by the automaton in Fig. 4.

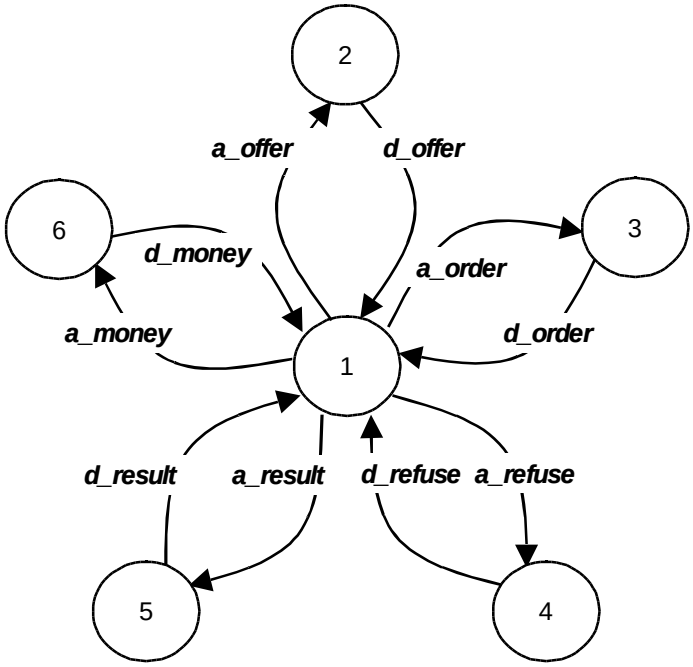

Fig. 4: Automaton of the communication system

This automaton defines a prefix closed language $C \subseteq \Sigma^*$ with $\Sigma = \{a\_offer, d\_offer, a\_order, d\_order, a\_refuse, d\_refuse, a\_result, d\_result, a\_money, d\_money\}$. The first letters "a" and "d" stand for "accept" or "deliver". The "effect" of F and G on the communication system is described formally by two alphabetic homomorphisms $\varphi:\Phi^* \to \Sigma^*$ and $\gamma:\Gamma^* \to \Sigma^*$. In our offer-order example, φ and γ are given by

$\varphi(r\_offer)=d\_offer$, $\varphi(s\_order)= a\_order$, $\varphi(s\_refuse)= a\_refuse$, $\varphi(r\_result)=d\_result$, $\varphi(s\_money)= a\_money$,

$\gamma(s\_offer)=a\_offer$, $\gamma(r\_order)= d\_order$, $\gamma(r\_refuse)= d\_refuse$, $\gamma(s\_result)=a\_result$, $\gamma(r\_money)= d\_money$

That is, a sending action ("s_...") of a cooperation partner corresponds to the acceptance of the message ("a_...") by the communication system and a receiving action ("r_...") of a cooperation partner corresponds to the delivery of the message ("d_...") by the communication system.

Due to these homomorphisms, $[\varphi,\gamma](EC) \subseteq C$ holds for the global system behavior where $[\varphi,\gamma]:(\Phi \cup \Gamma)^* \to \Sigma^*$ is the homomorphism defined by $[\varphi,\gamma](x)=\varphi(x)$ for $x \in \Phi$ and $[\varphi,\gamma](x)=\gamma(x)$ for $x \in \Gamma$. The three conditions $\pi_\Phi(EC) \subseteq F$, $\pi_\Gamma(EC) \subseteq G$ and $[\varphi,\gamma](EC) \subseteq C$ in conjunction now define the global system behavior completely:

$$EC = \pi_\Phi^{-1}(F) \cap \pi_\Gamma^{-1}(G) \cap [\varphi,\gamma]^{-1}(C).$$

In general, $[F,G]_c = \pi_\Phi^{-1}(F) \cap \pi_\Gamma^{-1}(G) \cap [\varphi,\gamma]^{-1}(C)$ is referred to as the *cooperation product* of F and G with respect to $c=(\Phi,\Gamma,\Sigma,\varphi,\gamma,C)$. That cooperation products are commutative as well as monotone in the component follows directly from the defintion.

For understanding the three components $\pi_\Phi^{-1}(F)$, $\pi_\Gamma^{-1}(G)$ and $[\varphi,\gamma]^{-1}(C)$ of a cooperation product, let us use the following four words as examples which lie each not within all three components and therefore not in their intersection either:

s_offer r_offer s_result s_order lies in $\pi_\Phi^{-1}(F)$, but not in $\pi_\Gamma^{-1}(G)$,
s_offer r_order r_offer s_money lies in $\pi_\Phi^{-1}(G)$, but not in $\pi_\Gamma^{-1}(F)$,
s_offer r_order r_offer s_order lies in $\pi_\Phi^{-1}(F) \cap \pi_\Gamma^{-1}(G)$, but not in $[\varphi,\gamma]^{-1}(C)$,
s_order r_order s_offer r_offer lies in $[\varphi,\gamma]^{-1}(C)$, but not in $\pi_\Phi^{-1}(F) \cap \pi_\Gamma^{-1}(G)$.

In contrast,

s_offer r_offer s_order r_order s_result lies in $\pi_\Phi^{-1}(F) \cap \pi_\Gamma^{-1}(G) \cap [\varphi,\gamma]^{-1}(C)$.

The following is to relate *realizations* (implementation specifications) of electronic contracts to requirements specifications by means of homomorphisms such that the "correct handling" of obligations is ensured in the implementation specifications.

We assume that the electronic contract EC is already available as a cooperation product, that is $EC=[F,G]_c$ with $c=(\Phi,\Gamma,\Sigma,\varphi,\gamma,C)$. That is no restriction because with C=EC, $F=\pi_\Phi(EC)$, $G=\pi_\Gamma(EC)$, $\Sigma=\Phi \cup \Gamma$ and φ and γ being equal to the identity on Φ and Γ, respectively, there is trivially such a representation of EC. Normally, however, there is a representation – as in our example - where C only describes the behavior of the communication system and contains no information about F and G, for example, in the case of Σ={*accept_msg*, *deliver_msg*} and C = (*accept_msg deliver_msg*)* (*accept_msg deliver_msg*)**accept_msg*.

Let $F' \subseteq \Phi'^*$ and $G' \subseteq \Gamma'^*$ be realizations of F and G with $\Phi' \cap \Gamma' = \emptyset$. The relations between F' and F, and between G' and G aredescribed by alphabetic homomorphisms $f:\Phi'^* \rightarrow \Phi^*$ and $g:\Gamma'^* \rightarrow G^*$, respectively. From formal point of view, a realization of a contracting party is thus a corresponding pair (F',f) or (G',g). That such a realization should be within the framework of the requirements specification, is expressed through the requirement

(9.1) $f(F') \subseteq F$ and $g(G') \subseteq G$

We assume for our further considerations that F' and G' use the same communication system as F and G and also handle the communication system "in the same way". That means: One may expect the correct working required by an "ideal contract" from the "real communication system". In formal terms, the global behavior EC' of the realization can be represented by a cooperation product $EC'=[F',G']_{c'}$ where the following holds:

(9.2) $c'=(\Phi',\Gamma',\Sigma,\varphi',\gamma',C)$ with $\varphi'=\varphi \circ f$ as well as $\gamma'=\gamma \circ g$.

The operation $\circ$ describes the composition of the homomorphisms, i.e. $\varphi'(x)=\varphi(f(x))$ for every $x \in \Phi'^*$ as well as $\gamma'(x)=\gamma(g(x))$ for every $x \in \Gamma'^*$.

A follow-up paper is to discuss the problem of realizing an ideal communication system as here assumed. This will show that a modular approach is justified: it allows to consider the realization of the contracting partners separately from the realization of the communication.

In [Och96]it has been shown that $[f,g]([F',G']_{c'})=[f(F'),g(G')]_c$ holds if (9.2) is true. Together with (9.1), hence $[f,g]([F',G']_{c'}) \subseteq [F,G]_c=EC$ follows. That is, the local inclusions (9.1) are also propagated to the global behavior. For the remainder of this section, (9.1) and (9.2) are to be assumed in general.

In addition to this global inclusion which describes a general safety property ("nothing wrong will happen" [AS85]), it is to be required for a correct realization that F' and G' are also able to fulfill their obligations: a specific continuation possibilities. For the formulation of a corresponding condition, the observation is useful that the ideal contract $EC \subseteq (\Phi \cup \Gamma)^*$ can be used

to describe the communication system between the local real partners F' and G' since EC contains the behavior of the (ideal) communication system C in a way compatible with F' and G' (9.2). In formal terms, this proposition reads as follows:

Theorem 1

If (9.1) and (9.2) are true, $[F',G']_{c'}=[F',G']_e$ with $e=(\Phi',\Gamma',\Phi\ \Gamma,f,g,EC)$ holds.That is: EC assumes the role of the communication system via f' and g' between F' or G'.

Proof:

According to the above, $[f,g](x)$ EC holds for every x $[F',G']_{c'}$ Since $\pi_{\Phi'}(x)$ F' and $\pi_{\Gamma'}(x)$ G also hold for these x by definition, x $[F',G']_e$ follows, i.e. $[F',G']_{c'}$ $[F',G']_e$.

$[f,g](y)$ $EC=[F,G]_c$ holds for y $[F',G']_e$. Hence $[\varphi',\gamma'](y)=[\varphi,\gamma]([f,g](y))$ $[\varphi,\gamma]([F,G]_c)$ C follows. Since $\pi_{\Phi'}(y)$ F' and $\pi_{\Gamma'}(y)$ G' also hold for these y by definition, y $[F',G']_{c'}$ follows, i.e. $[F',G']_e$ $[F',G']_{c'}$.

The benefit of the above is the fact that, under the given conditions, the cooperation form c or c' does no longer occur explicitly. It was only used to formulate condition (9.2). So to speak, one once formulates C Σ^*, then the communication rules of EC (ideal contract)" are given as communication system" for every implementation F' and G'.

In the case of realizations (F',f) and (G',g), the homomorphisms f and g transfer the obligations of F and G to F' and G'. We say: *F' is under the obligation M after a sequence of actions* $y\in[F',G']_e$ if there is a decomposition y=uv with $[f,g](v)$ V and $(\pi_\Phi([f,g](v)),M)$ O_F; this applies accordingly to G' and to the communication system. If a z $y^{-1}([F',G']_e)\cap\Phi'^*$ with f(z) M exists, then we say *F' fulfills its obligations M with a sequence of actions* $yz\in[F',G']_e$; this applies accordingly to G'. With respect to the communication system K, the corresponding definition is somewhat different since we have assumed that the realization uses the same communication system as the requirements specification: *K fulfills its obligations M with the sequence of actions* $yz\in[F',G']_e$, if a z $y^{-1}([F',G']_e)\cap(\Phi'\ \Gamma')$ with $[f,g](z)$ M exists.

Since the concrete realization of a contracting party is in general unknown to the other contracting party, the conditions still to be defined have to ensure with respect to the obligations that both parties can fulfill their obligations in the interaction of any "correct" realization of G with any "correct" realization of F. That is, these correctness conditions have to be formulated separately for the realizations of the partners. The interaction of the realization of one party with the requirements specification of the other party is considered for this purpose. For reasons of symmetry, it suffices to discuss one of the two cases.

Definition (obligation-conforming realization):

Let $d=(\Phi',\Gamma',\Phi\ \Gamma,f,i_\Gamma,EC)$ where i_Γ is the identity homomorphism on Γ^*. The cooperation product $[F',G]_d$ then describes the mentioned interaction of F' and G. A realization (F',f) is *obligation-conforming* if conditions (9.3) and (9.4) are fulfilled:

(9.3) For every u' $[F',G]_d$ and v' $u'^{-1}([F',G]_d)$ with $[f,i_\Gamma](v')$ V and $(\pi_\Phi([f,i_\Gamma](v')),M)$ O_F, there is a z' $(u'v')^{-1}([F',G]_d)\cap\Phi'^*$ with f(z') M.

(9.4) For every u'$[F',G]_d$ and v' $u'^{-}1([F',G]_d)$ with $[f,i_\Gamma](v')$ V and $(\pi W([f,i_\Gamma](v')),M)$ O_K, $f((u'v')^{-1}([F',G]_d)\cap\Phi')\cap\Phi^0=([f,i_\Gamma](u'v'))^{-1}(V)\cap\Phi^0$ holds.

Interpretation of (9.3): If a real behavior u'v' of F' creates an obligation of F in accordance with EC, then a real sequence of actions which is regarded as a fulfillment M of F' according to EC (f(z') M) is available for F'.

Interpretation of (9.4): If a real sequence of actions u'v' creates an obligation for the communication system to act with respect to F in Φ^0, e.g. to deliver a message to F', then any fulfillment of the obligation by the communication system with respect to F (in Φ^0) is actually implemted (in Φ'); in our example: F' can accept the message.

The next theorem says that the obligation conformity of realizations yields the desired effect.

Theorem 2

If (F',f) and (G',g) are obligation-conforming realizations, the following propositions (95.) to (9.7) are true for F', G' and K:

(9.5) For every u $[F',G']_e$ and v $u^{-1}([F',G']_e)$ with $[f,g](v)$ V and $(\pi_\Phi([f,g](v)),M)$ $O_{F,}$ there is a z $(uv)^{-1}([F',G']_e)\cap\Phi'^*$ mit f(z) M.

(9.6) For every u $[F',G']_e$ and v $u^{-1}([F',G']_e)$ mit $[f,g](v)$ V und $(\pi_\Gamma([f,g](v)),M)$ $O_{G,}$ there is a z $(uv)^{-1}([F',G']_e)\cap\Gamma'^*$ with g(z) M.

(9.7) For every u $[F',G']_e$ and v $u^{-1}([F',G']_e)$ with $[f,g](v)$ V and $(\pi_\Omega([f,g](v)),M)$ O_K, $[f,g]((uv)^{-1}([F',G']_e\cap)(F\ 'G'))\cap\Omega = ([f,g](uv))^{-1}(V)\cap M$ holds.

Interpretation of (9.5): If F' creates an obligation, the satisfiability by F is actually implemented in F'. Beyond (9.3), this shows that it is also true in the interaction with the real G'. As the proof is to show, only the communication conformity (9.2) is required from the other side G', not the obligation conformity (9.3).

Interpretation of (9.6) – analogously to (9.5) for G'Interpretation of (9.7): The fulfillment of the obligations of the communication system to F and G (e.g. delivery of messages) is actually implemented in F' and G', i.e., it can actually be accepted by the real F' and G'.

Proof:

For u $[F',G']_e$ and v $u^{-1}([F',G']_e)$ with $[f,g](v)$ V and $(\pi_\Phi([f,g](v)),M)$ O_F the following holds:

$[i_{\Phi'},g](u)$ $[F',G]_d$,
$[i_{\Phi'},g](v)$ $([i_{\Phi'},g](u))^{-1}([F',G]_d)$,
$[f,i_\Gamma]([i_{\Phi'},g](v))$ V and
$(\pi_\Phi([f,i_\Gamma]([i_{\Phi'},g](v))),M)$ O_F .

Because of (9.3), there is a z $([i_{\Phi'},g](u)[i_{\Phi'},g](v))^{-1}([F',G]_d)\cap\Phi'^*$ with f(z) M. Hence $[i_{\Phi'},g](u)[i_{\Phi'},g](v)z$ $[F',G]_d$ follows and therefore $\pi_{\Phi'}(uvz)=\pi_{\Phi'}([i_{\Phi'},g](u)[i_{\Phi'},g](v)z)$ F'. Because of z Φ'^*, hence $[f,g](uvz)=[f,i_\Gamma]([i_{\Phi'},g](u)[i_{\Phi'},g](v)z)$ EC as well as $\pi_{\Gamma'}(uvz)=\pi_{\Gamma'}(uv)$ G follow. Together, that means uvz $[F',G']_{e,}$ i.e., z $(uv)^{-1}([F',G']_e)\cap\Phi'^*$. In this way, (9.5) has been proved; (9.6) can be proved accordingly.

By means of analogous considerations and the disjoint decomposition $\Omega = \Phi^0\ \Gamma^0$, (9.7) can be derived from (9.4) and from a corresponding property of G'.

It is to be noted that theorem 2 ensures continuation possibilities only for the sequences of actions from $[F',G']_e$, after which F', G' or K are under an obligation. If one wants to guarantee continuation possibilities according to EC for all sequences of actions from $[F',G']_e$, then the *simplicity* of [f,g] has to be required [Och94]; [Och96] specifies criteria for this purpose. The proposition of theorem 2, however, does not follow from this simplicity vice versa, since it ensures the existence of quite specific continuation possibilities related to the satisfiability of

obligations which is not guaranteed, however, by the simplicity in this respect to the obligations.

Since, because of theorem 2, obligation-conforming realizations do not hinder the fulfillment of obligations, the pressuring to goal is transferred to such realizations. The proofs which have to be collected by realizations (see Section 13 for further details about proofs) are then elements of $\pi_\Phi([f,g]([F',G']_e))$ for F', $\pi_\Gamma([f,g]([F',G']_e)$ for G' and $\pi_\Omega([f,g]([F',G']_e)$ for K.

10 Obligations of the communication system

Obligations refer to a cooperation partner or the communication system. Its formal definition, however, has to refer to the global behavior in the binding phase V, since a reasonable definition of obligations should consider their satisfiability. With respect to the communication system, however, this has the effect that the obligations can depend on the state of cooperation which is certainly not desirable since the communication system is only a means to an end and does not know the actual state of cooperation.

The communication system need not reconstruct – as before - the sequence of actions of the partners. A communication specific order suffices, for example, "*deliver_order* after *accept_order*" (regardless of whether any *offer* actions occurred before) or FIFO or LIFO rules.

The use of cooperation products in the formalization allows the obligations of the communication system to be independent of the state of cooperation. For distinguishing the concept to be defined now from the definition in Section 7, we will call it *communication obligation.*

Let $EC=[F,G]_c$ with $c=(\Phi,\Gamma,\Sigma,\varphi,\gamma,C)$. In Section 7, the alphabet of the communication system has been denoted by Ω. It contains all letters of the contracting parties Φ and Γ but no letters "of its own". Now, the communication system should be provided with letters of its own which describe independent actions abstracted from a concrete contract. We denote this alphabet by Σ. Later we will show how these two views are interrelated, i.e., how the abstract actions of the communication system from Σ are expressed in a concrete cooperation from Ω.

Like in Section 7, our considerations are based on a disjoint union $\Sigma=\Sigma \dot\cup \Sigma^0$ where Σ contains the actions which are in the responsibility of the communication system, i.e., typically actions such as the delivery of messages to the corresponding recipient. Normally, these are also actions the communication system is obliged to perform. However, it is also possible that the communication system is obliged to do less, namely if only the delivery of messages of a specified transmission type is guaranteed.

(10.1) Let $\Sigma \subseteq \Sigma$ be the set of these actions the communication system is obliged to perform. A *communication obligation* is defined as a pair $(z,N) \in C \times \mathcal{P}(\Sigma)$ with $N=z^{-1}(C)\cap\Sigma \neq \emptyset$.

Interpretation of (10.1): Among that what can occur after z is also an action from Σ. All this is a fulfillment of an obligation. N will often consist of only one letter, e.g. *(z,N)=(accept_order, {deliver_order})*. However, N can also consist of several action alternatives, e.g. *(z,N)=(accept_order, {deliver_order, send_nonDeliveryNotification})*.

The set of these communication obligations only depends on C and Σ, it is denoted by O_C. These are the "movement expressions" from [Gri94] which we now recognize as abstract obligations of the communication system. Therefore, we do not need any new description element (as in [Gri94]), but we use the already introduced description means of *obligation* for describing the *movements of messages* from one partner through the communication system to the other partner! For differentiating obligations of a communication system within a concrete

cooperation, we want to refer to these abstract communication obligations nonetheless as *movement expressions*.

Using the homomorphisms φ and γ, the obligations of the communication system K in the sense of Section 7 (i.e., obligations with respect to every concrete sequence of actions in F or G) can be derived from the movement expressions (i.e., the abstract communication obligations). For this purpose, we set

(10.2) Ω =[φ,γ]$^{-1}$(Σ)∩(Φ Γ) and Ω^0=[φ,γ]$^{-1}$(Σ^0)∩(Φ Γ) .

From Ω =Φ^0 Γ^0, Φ=Φ Φ^0 and Γ=Γ Γ^0 as well as from Φ∩Γ=Φ ∩Φ^0=Γ ∩Γ ^{0}= we obtain

(10.3) Φ^0=Φ∩Ω and Γ^0=Γ∩Ω as well as Φ =Φ\$\Phi^0$ und Γ =Γ\$\Gamma^0$.

Then, for the internal actions of F and G mentioned in Section 7, the following holds

(10.4) Φ \$\Omega^0$=Φ∩φ^{-1}({ε}) resp. Γ \$\Omega^0$=Γ∩γ^{-1}({ε}).

The set O_K of obligations of the communication system in the concrete cooperation EC=$[F,G]_C$ is then defined by means of the movement expressions O_C in the sense of (10.1) as follows:

(10.5) (x,M) O_K if and only if (x,M) π_Ω(EC)× (Ω) which fulfills the conditions of (7.1) and (7.2) and if there is a N Σ with ([φ,γ](x),N) O_C .

With this definition, a promise of the communication system x would not necessarily be in a binding phase V of a contract EC, but could be located outside, e.g. before the binding phase. This is reasonable for a communication system since the contracting parties must also rely outside the binding phase on a communication safely arriving at the other side.

However, for pressuring to the goal of a cooperation contract, only the binding phase V is of importance. Only in this phase, the partners collect proofs and must be able to conclude a promise of the other partner from these proofs by means of the movement expressions. Therefore, it is reasonable to consider the obligations of the communication system O_K only within the binding phase. Since a binding phase V is not simply a subset of its associated contract EC, but consists of segments of sequences of actions in EC, such a view restricted to V has to be expressed in a roundabout way by means of prefixes:

The set O_K of the obligations of the communication system within a binding phase V of the concrete cooperation EC=$[F,G]_C$ consists of these elements (x,M) π_Ω(V)× (Ω):

(10.6) (x,M) O_K if and only if (x,M) π_Ω(V)× (Ω) which fulfills the conditions (7.1) and (7.2) and if there is a y $[F,G]_c$ and a N Σ with
x π_Ω(y^{-1}($[F,G]_c$)∩V) and ([φ,γ](yx),N) O_C .

M [φ,γ]$^{-1}$(Σ)∩(Φ Γ) then follows from the definition of the cooperation product. With (10.1), (10.2) and (10.5), O_K is completely defined by the selection Σ Σ Σ as well as by φ, γ, V and C.

Unlike the "obligation of a communication system" from Section 7 which was formulated there as a component of a concrete cooperation, the new concept of "communication obligation" (or "movement expression") in the present section is independent of an imbedding in a concrete cooperation: it uses the alphabet of the communication system (10.1) which is independent of the cooperation partners. The two views are mapped onto each other by a homomorphism (10.2-5). In this way, we have got a conceptual apparatus to describe special communication systems. The example of a communication system showing 6 states in Section 9 (Fig. 4) realizes a simple send-receive mechanism between the communication partners. The

aim to specify a communication system that delivers an entered message regardless of its type (*offer*, *order*, *result* etc.) in a finite number of steps, requires further definitions to be developed in a follow-up paper.

11 Multiple runs through a binding phase

As mentioned above (Section 5), the binding phase V can be passed repeatedly in an electronic contract EC. The contracting parties must be able to recognize when they enter a binding phase and in which phase they presently are. To express this, some additional requirements for V and EC are to be formulated in this section.

With respect to F, the *local enter actions* into a binding phase are the elements of $\pi_\Phi(V) \cap \Phi$. To signal the entry into V in any situation for F, they must fulfill the following condition:

(11.1) To every u EC$\cap$(Φ Γ)*(π_Φ(V)$\cap\Phi$) there are w,v (Φ Γ)* with u=wv, v V and π_Φ(v) Φ.

That means: a letter c π_Φ(V) $\cap\Phi$ which looks for F as if belonging to the binding phase, actually has to be part of a sequence of actions v in the binding phase: π_Φ(v)=c.

The corresponding conditions must also be fulfilled with respect to G and K.

Individual actions performed by the partners, such as receiving of a specific message, can be proved non-repudiatably by electronic signatures. More difficulties will arise from a sequence of actions which is in general more than the set of its individual actions. For this purpose, let us have a look at the mapping alph:$\Phi^* \rightarrow$ (Φ) which assigns the set of letters occurring in a word to this word. The restriction of this mapping to π_Φ(V) is denoted by a_F; the following is to hold for it:

(11.2) a_F:π_Φ(V)$\rightarrow$ (Φ) is injective and x pre(y) holds for all x,y π_Φ(V) with a_F(x) a_F(y).

Injectivity means that a sequence of actions in V must not occur again in V in another order. In addition, the same actions must not occur multiply within an obligation expression. The corresponding conditions must also be fulfilled with respect to G and K. (11.2) ensures that a set of proofs for *individual actions* is also a unique proof for a specific *sequence of actions as a whole*.

Since a binding phase V can be passed repeatedly within EC, it is necessary to assign the proofs uniquely to a run. An "unfolded version" EC_τ of EC is used for that purpose.

To represent the unfolding of sequences of actions in several runs, one extends the alphabets, for example by run indices, and then ensures by a restriction of the possible words that the rounds have to be passed completely before beginning with the next round.

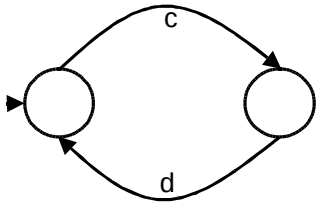

Fig. 5: Example of an automaton for a binding phase "first c then d"

Example: Let $\Delta=\{c,d\}$ be an alphabet over the two actions *c* and *d* which may be executed in a language D only in the order "first *c* then *d*". Several runs would then be expressed through the words $D = \{\varepsilon, c, cd, cdc, cdcd, cdcdc, ...\}$. To distinguish the various rounds from each other (first *c* then *d*) the actions can be indexed with the round number $\Delta_\tau=\Delta\times\mathbb{N}= \{c_1, d_1, c_2, d_2, c_3, d_3, c_4, ...\}$.. One now requires from the associated sequences of actions "accuracy of rounds" as a restriction: $D_\tau=\{\varepsilon, c_1, c_1d_1, c_1d_1c_2, c_1d_1c_2d_2, c_1d_1c_2d_2c_3, ...\}$. That is: all rounds have to be executed completely before beginning with the next round, no rounds may be left out. Sequences like $c_2d_1c_5d_3c_1$ (round mix) or $c_1d_1c_3d_3$ (round 2 was forgotten) are explicitly inadmissible even if they comply with the rule "first *c* then *d*". The indexed alphabet Δ_τ is infinite and therefore cannot be mapped bijectively onto the two-element Δ. Though the "projection function" $\tau: \Delta_\tau^* \rightarrow \Delta^*$ that suppresses the indices, is no bijection on the *alphabets*, their restriction to the language D_τ is indeed a bijection between the *languages* D_τ and D: a $c_1d_1c_2d_2c_3$.. ... D_τ uniquely corresponds to a *cdcdc*... D. The reason for that is the fact that D_τ has been restricted such that the round indices change only after completed rounds and may only ascend contiguously.

For example, the various runs of a binding phase $V=\{\varepsilon, c, cd\}$ through an electronic contract $EC=\{\varepsilon, a, ab, abc, abcd, abcdc, abcdcd,...\}$could be identified by the following indexing: $EC=\{\varepsilon, a, ab, abc_1, abc_1d_1, abc_1d_1c_2, abc_1d_1c_2d_2,...\}$.

In general, one proceeds as follows: Let Φ_τ and Γ_τ be two disjoint sets which represent the "unfolded" alphabets Φ and Γ . Accordingly, let EC_τ $(\Phi_\tau \Gamma_\tau)^*$ be the "unfolded" electronic contract which distinguishes several runs through a binding phase from each other. The "projection function" $\tau:(\Phi_\tau \Gamma_\tau)^* \rightarrow (\Phi \Gamma)^*$ (which suppresses the round indices) is an alphabetic homomorphism with $\tau(\Phi_\tau)=\Phi$ and $\tau(\Gamma_\tau)=\Gamma$. Let the restriction of τ to EC_τ be denoted by τ_E. The following has then to hold for EC_τ and τ_E:

(11.3) $\tau_E(EC_\tau)= EC$,

(11.4) $\tau_E:EC_\tau \rightarrow EC$ is a bijection and

(11.5) $alph(u)\cap alph(v)=$ holds for uv EC_τ with $\tau(v)$ V.

Once again: $\tau_E:EC_\tau \rightarrow EC$ is a bijection of the languages EC_τ and EC but τ½$_{\Phi\tau\ \Gamma\tau}:\Phi_\tau \Gamma_\tau \rightarrow \Phi \Gamma$ is in nearly all cases no bijection of the alphabets.

For a realization $[F',G']_{c'}$ of $EC=[F,G]_c$ (Section 9), the following is to be required:

(11.6) Alphabetic homomorphisms $f_\tau:\Phi'^* \rightarrow \Phi_\tau^*$ and $g_\tau:\Gamma'^* \rightarrow G_\tau^*$ with $[f,g]=\tau o[f_\tau,g_\tau]$ exist (i.e., the realizations themselves maintain identifiers).

For the unique allocation of the proofs to a run, we must be able to determine for every action set A Φ' Γ' whether all its actions belong tothe same run of V, i.e. whether there is a uv EC_τ with $\tau(v)$ V and $[f_\tau,g_\tau](A)$ $alph(v)$.

It is reasonable not to demand directly the judicial capability of being proved for the distinction between "new" and "old" identifiers since this would require from every partner to save his complete history of actions. This can be avoided if every partner accepts the selected identifier for the new run together with the corresponding local enter action into V. Since a refusal should also be feasible here, a local enter action must not involve any obligations.

12 Overlapping runs through a binding phase

The following shortened version of the offer-order-deliver-pay cooperation with which payment is made upon the placement of an order *(order*: order and payment), shows that the conditions used so far are still too restrictive.

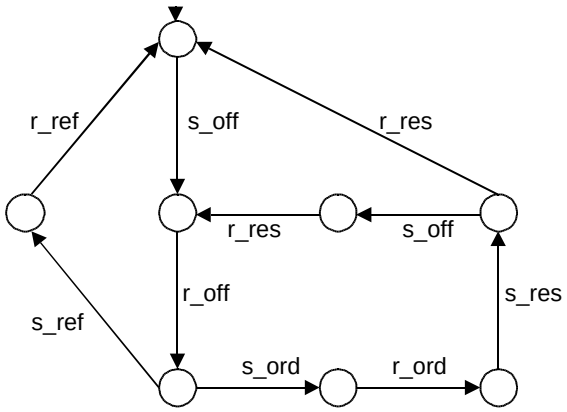

Fig. 6: Example for EC with overlapping binding phases: "first r_res then s_off" or vice versa

If we assume that the binding phase begins here with *s_off* (offer) and ends with *r_ref* (refusal) or *r_res* (result: delivery of goods) , V consists of the following words:

V = pre({ *s_off r_off s_ref r_ref* , *s_off r_res r_off s_ref r_ref* ,

s_off r_off s_ord r_ord s_res r_res , *s_off r_off s_ord r_ord s_res s_off r_res* ,

s_off r_res r_off s_ord r_ord s_res r_res , *s_off r_res r_off s_ord r_ord s_res s_off r_res* }).

Unlike our first example, this example shows that not every complete run through the binding phase is terminated by the partner who started it. That is illustrated in Fig. 6 on the upper right state: The follow-up offer *s_off* can be dispatched both before or after receiving the delivery of goods for the current order *r_res*. In general, $x^{-1}(EC) \cap (\Phi \quad \Gamma) = z^{-1}(V) \cap (\Phi \quad \Gamma)$ no longer holds for x EC with x=yz and z V\(max(V) {ε}). Instead of this condition only

Closure: x EC with x=yz and z V\(max(V) {ε}) holds:
$x^{-1}(EC) \cap (\Phi \quad \Gamma) \quad z^{-1}(V) \cap (\Phi \quad \Gamma)$

can now be required. This weakening has effects on the satisfiability of obligations. For this, a consistent system of definitions and conditions is now specified. The technique of unfolding as described in Section 11 is here of primary importance to distinguish "relevant" actions in a binding phase from "irrelevant" actions.

If F is now under obligation M after a sequence of actions y V, the definition of the obligation only guarantees that a follow-up action from Φ will occur in V, but not necessarily in EC. To maintain the pressuring to goal, the *satisfiability of obligations* has to be required in addition:

(12.2) O_F is *satisfiable in EC*, if for every z EC and every (x,M) O_F with z=uy and $x=\pi_\Phi(y)$ the property $z^{-1}(EC) \cap \Phi \neq$ holds.

The corresponding conditions must also be fulfilled for O_G and O_K.

When F performs the action *r_res* after several runs through V, he cannot learn by that whether G has already started a new run or not. Therefore *r_res* cannot be a local enter action of F and must not be relevant to obligations of F in a new run. The weakened definition of the binding phase demands the definition of *relevant actions* for obligations and local enter actions into a binding phase.

Let Φ_ρ and Γ_ρ be two disjoint sets, let V_ρ $(\Phi_\rho \cup \Gamma_\rho)^*$ and $\rho:(\Phi_\rho \cup \Gamma_\rho)^* \rightarrow (\Phi \cup \Gamma)^*$ be an alphabetic homomorphism with $\rho(\Phi_\rho)=\Phi$ and $\rho(\Gamma_\rho)=\Gamma$. We require for it that its restrictions to V_ρ, $\pi_{\Phi\rho}(V_\rho)$ and $\pi_{\Gamma\rho}(V_\rho)$ define bijections $\rho_V:V_\rho \rightarrow V$, $\rho_F:\pi_{\Phi\rho}(V_\rho) \rightarrow \pi_\Phi(V)$ and $\rho_G:\pi_{\Gamma\rho}(V_\rho) \rightarrow \pi_\Gamma(V)$. The irrelevant actions are identified by means of their complements Φ_ρ and Γ_ρ. $\Phi_\rho \cap \Phi$ or $\Gamma_\rho \cap \Gamma$ are the *relevant actions* of F or G. In addition, let $\rho(x)=x$ for every $x \in (\Phi_\rho \cap \Phi) \cup (\Gamma_\rho \cap \Gamma)$.

To ensure that obligations depend only on the relevant actions, we require:

(12.3) If $(x,M) \in O_F$ and $x' \in \pi_\Phi(V)$ with $\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))=\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x'))$, then $M' \subseteq \Phi$ with $(x',M') \in O_F$ exists.

That is, whether there is an obligation in $x \in \pi_\Phi(V)$, does not depend on x, but only on $\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))$. A corresponding condition must also be fulfilled with respect to G. $\pi_{\Phi\rho\cap\Phi}(V_\rho)$ is the *set of the relevant sequences of actions* of F with respect to obligations. Accordingly, the proofs of F are also elements of $\pi_{\Phi\rho\cap\Phi}(V_\rho)$. The conditions (11.1) and (11.2) are then to be replaced by the following:

(12.4) For every $u \in EC \cap (\Phi \cup \Gamma)^*(\pi_{\Phi\rho\cap\Phi}(V_\rho) \cap \Phi)$ there are $w,v \in (\Phi \cup \Gamma)^*$ with $u=wv$, $v \in V$ and $\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(\pi_\Phi(v))) \subseteq \Phi$.

(12.5) $a_F:\pi_{\Phi\rho\cap\Phi}(V_\rho) \rightarrow \Sigma(\Phi)$ is injective, and $x \in pre(y)$ holds for all $x,y \in \pi_{\Phi\rho\cap\Phi}(V_\rho)$ with $a_F(x) \in a_F(y)$.

The local enter actions of F are now the elemnts of $\pi_{\Phi\rho\cap\Phi}(V_\rho) \cap \Phi$. a_F here denote the restriction of the mapping alph onto $\pi_{\Phi\rho\cap\Phi}(V_\rho)$. To identify at local level with continuing cooperation which of the performed actions makes a contribution to an element from $\pi_{\Phi\rho\cap\Phi}(V_\rho)$, the following is required in addition:

(12.6) For $xa \in \pi_\Phi(V)$ with $a \in \Phi$ the following holds:
$\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(xa)) = \pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))a$, if $a \in \pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))^{-1}(\pi_{\Phi\rho\cap\Phi}(V_\rho))$, and
$\pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(xa)) = \pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))$, if $a \notin \pi_{\Phi\rho\cap\Phi}(\rho_F^{-1}(x))^{-1}(\pi_{\Phi\rho\cap\Phi}(V_\rho))$.

The condition (11.5) is still to be adapted to the relevant actions; it is replaced by:

(12.7) For $uv \in EC_\tau$ with $\tau(v) \in V$ the following holds:
$alph(u) \cap alph(v) \cap \tau^{-1}(alph(\pi_{(\Phi\rho\cap\Phi) \cup (\Gamma\rho\cap\Gamma)}(\rho_V^{-1}(\tau(v))))) = \emptyset$.

13 Proofs

As already mentioned above (Section 8), for solving conflicts, i.e. if the cooperation does not continue as planned, every cooperation partner and the communication system have to be able to prove non-repudiatably which actions have taken place on his/its side or which actions he/it may expect from the other side. Therefore, *proofs* must be collected; in formal terms, these are elements from $\pi_\Phi(V)$ for F, $\pi_\Gamma(V)$ for G and $\pi_\Omega(V)$ for K.

In the following we assume that all necessary actions are actually provable and are therefore non-repudiatable. For receiving actions of receipt, this is feasible by means of a digital signature by the partner. For sending actions, it is feasible with the aid of receipts given by the partner or the communication system.

We take the point of view of F who performed the sequence of actions x $\pi_\Phi(V)$ and show how F can proceed: F puts his locally performed sequence of actions x (together with the associated proofs) on the table and requires from K and G to do the same; let K disclose v and let G disclose w (the same argument can be extended to as many partners as desired).

$D := \pi_\Phi^{-1}(x) \cap \pi_\Omega^{-1}(v) \cap \pi_\Gamma^{-1}(w) \cap V$ consists of all sequences of actions which fit the submitted local observations and are therefore non-repudiatable.

If D= , then the submitted proof for x, v and w are inconsistent and have to be verified by an independent third party. That can happen in two cases:

1. if a participant submits a false (and having therefore no evidentiary value) local sequence of actions - that is clarified by the other partners on account of the body of evidence;
2. because a partner submits a too short local sequence of actions ("I did not receive or perform the last segment") - that is refuted by a non-repudiatability proof of another partner.

If D≠ , then, from a global point of view, a sequence of actions from the intersection D has been performed and the same situation of obligation exists in all sequences of actions from D since according to (7.1) an obligation only depends on the locally visible events which here are not identical for every element in D. This forces the continuation of cooperation if at least one partner is under an obligation. If no partner is under an obligation, max(V) or no single subgoal is achieved due to coverage (8.1), i.e., success linkage (6.1) is preserved.

From a global point of view, this means nothing else than that every participant in a cooperation has to collect these two types of evidence: his view of the obligation preconditions of all other participants ("you are obliged to do something") as well as his own obligation fulfillment ("I am no longer under any obligation"). That corresponds to the "balance condition" in [Gri94].

The procedure described here helps every participant who behaves well to reach the goal - even if his partners do not cooperate – by following this action maxim, as soon as someone may have achieved a first subgoal:

1. within a binding phase, never act if you are not under an obligation;
2. if cooperation makes no progress, fulfill your own obligations first;
3. if cooperation makes no progress and you are under no obligation, disclose your proofs and demand the same from all your partners thus forcing the obliged partner to act;
4. if no partner is under any obligation, then, due to coverage (8.1), either all goals have been achieved or no subgoals have been achieved as yet: success linkage (6.1) is preserved.

However, the procedure is somewhat unpleasant since, in a no-progress situation, a participant will possibly shrink back from arguing with all partners at once. Therefore, it is reasonable to specify an electronic contract such that every partner has a *strategy for the sequential processing of the obliged partners* in any situation. The simplest way to do that is to have all (finitely many!) y $\pi_\Phi^{-1}(\{x\}) \cap V$ ordered strictly monotonously by ascending length. This is the case in our simple offer-order-deliver-pay cooperation, for example. F can then process the y in this order. A partner who is confronted with such a y either fulfills the relevant obligations thus advancing cooperation again or he proves that he has already fulfilled the relevant obligation thus leading up to the next longer element y.

It would even suffice that the y can be ordered in a semi-ordered way (tree-like). The branch-points of such a tree mean unconditional (however alternative "or"-ed) obligations for the fulfillment of which only *one* partner is responsible in any case. Such a strategy is the core of the balance model.

14 Conclusion

In an open network, one cannot have control of the correct behavior of the other partner. With the preconditions elaborated here, one is however protected against a behavior of remote cooperation partners which is not in conformity with the contract. The obligation-conforming local realization of an electronic contract (Section 9) and the proper handling of proofs (Section 13) ensure the pressuring to goal for a real cooperation even without having to require globally the correctness of implementation on the other side. It suffices to have a correct local implementation and to behave correctly.

This paper formalizes how obligations and proofs have to play together for achieving pressuring to goal. It does not formalize what proofs are; this will be done in a forthcoming paper.

For pressuring to goal, we need not demand that the other partners fulfill their obligations voluntarily, if we presuppose that the proofs are of non-repudiatable validity with respect to third parties who are also able to *enforce* the progress of a cooperation on the basis of a proved and non-fulfilled promise! Based on this precondition, the following is true: either a partner keeps his promise because he is good-natured, or he is forced to do so by third parties (courts) on the basis of proofs. In both cases, cooperation is *really* completed in max(V) by which both partners or none of the partners achieve their goal because of the success linkage. And therefore a participant is protected in his goals even if his partner does not cooperate whether for viciousness or because his implementation is incorrect.

This is the type of security we need in open networks which cannot be controlled globally. The formalization can be mapped to multilateral cooperation without problems.

15 References

[AS85] Alpern, Bowen and Schneider, Fred B.: Defining Liveness. Information Processing Letters 24, 1985.

[Eil74] Eilenberg, Samuel: Automata, Languages and Machines, Vol. A . Academic Press, New York, 1974, 451 S.

[Gri94] Grimm, Rüdiger: Sicherheit für offene Kommunikation - Verbindliche Telekooperation. B.I. Wissenschaftsverlag, Mannheim, 1994, 274 S.

[GO99] Grimm, Rüdiger und Ochsenschläger, Peter: Verbindliche Telekooperation - ein Modell für Electronic Commerce auf der Basis formaler Sprachen. In: Röhm, Fox, Grimm, Schoder: Sicherheit und Electronic Commerce. DuD-Fachbeiträge, Vieweg, Wiesbaden. September 1999, S. 1-14.

[Och94] Ochsenschläger, Peter: Verification of Cooperating Systems by Simple Homomorphisms. Workshop Algorithmen und Werkzeuge für Petrinetze Berlin 1994

[Och96] Ochsenschläger, Peter: Kooperationsprodukte formaler Sprachen und schlichte Homomorphismen. Arbeitspapiere der GMD 1029. Sankt Augustin, November 1996, 52 S.