Protune Negotiation Model

J. L. De Coi, D. Ghita, D. Olmedilla

December 4, 2006

This document aims at formalizing the negotiation between two entities, namely E_1 and E_2. For the rest of the paper we assume that E_1 is the initial requester while E_2 is the provider, i.e. E_1 is assumed to send a request to E_2 thus starting a negotiation. Notice that during a negotiation both entities E_1 and E_2 may both request or provide information to each other.

Let our Policy language be a rule-based language. Such a rule language is based on normal logic program rules $A \leftarrow L_1, \ldots L_n$ where A is a standard logical atom (called the *head* of the rule) and $L_1, \ldots L_n$ (the *body* of the rule) are literals, i.e. L_i equals either B_i or $\neg B_i$, for some logical atom B_i.

Definition 1 (Policy) *A Policy is a set of rules, such that negation is not applied to any predicate occurring in a rule head.*

This restriction ensures that policies are *monotonic* in the sense of [**?**], i.e. as more credentials are released and more actions executed, the set of permissions does not decrease. Moreover, the restriction on negation makes policies *stratified programs*; therefore negation as failure has a clear, PTIME computable semantics that can be equivalently formulated as the perfect model semantics, the well-founded semantics or the stable model semantics [**?**].

Definition 2 (Negotiation Message) *A Negotiation Message is an ordered pair*

$$(p, C)$$

where

- $p \equiv$ *a Policy*
- $C \equiv$ *a set of credentials*

We will denote with M the set of all possible Negotiation Messages.

Definition 3 (Message Sequence) *A Message Sequence σ is a list of Negotiation Messages*

$$\sigma_1, \ldots \sigma_n \mid \sigma_i \in M$$

We will denote with $|\sigma|$ the length of σ and with σ_i the i-th element of the Message Sequence σ.

Definition 4 (Negotiation History) *Let E_1 and E_2 be the two entities involved in the negotiation. Let E_1 be the initiator of such a negotiation, i.e. the sender of the first message in the negotiation. A Negotiation History σ for the entity E_j $(j = 1, 2)$ is a Message Sequence*

$$\sigma_1, \ldots \sigma_n \mid \sigma_i \in M$$

Moreover let

- $M_{snt}(\sigma) = \{\sigma_i \mid i = 2k - (j \ mod 2), 1 \leq k \leq \lfloor n/2 \rfloor\}$
- $M_{rvd}(\sigma) = \{\sigma_i \mid i = 2k - 1 + (j \ mod 2), 1 \leq k \leq \lfloor n/2 \rfloor\}$

We will refer to a Negotiation History also as a *Negotiation State*.

Intuitively, messages among parties are sent alternatively, i.e. a message sent by E_j is followed by the reception of a message, which is in turn followed by the sending of a new message and so on. Therefore, $M_{snt}(\sigma)$ (resp. $M_{rvd}(\sigma)$) represents the set of messages sent (resp. received) by E_j.

Notice that according to this definition, the Negotiation History σ is shared by the two entities E_1 and E_2, but the sets $M_{snt}(\sigma)$ and $M_{rvd}(\sigma)$ are swapped among them. Therefore it holds that

$$M_{snt}(E_1, \sigma) = M_{rvd}(E_2, \sigma)$$

and

$$M_{rvd}(E_1, \sigma) = M_{snt}(E_2, \sigma)$$

In order to ease the notation in the rest of the paper, given a Negotiation History σ, we define the following entities

- $C_{snt}(\sigma) = \bigcup\{C_i \mid \exists p_i \ (p_i, C_i) \in M_{snt}(\sigma)\}$
- $C_{rvd}(\sigma) = \bigcup\{C_i \mid \exists p_i \ (p_i, C_i) \in M_{rvd}(\sigma)\}$
- $lp_{snt}(\sigma) = p_{i_{max}} \mid i_{max} = max\{i \mid (p_i, c_i) \in M_{snt}(\sigma)\}$
- $lp_{rcv}(\sigma) = p_{i_{max}} \mid i_{max} = max\{i \mid (p_i, c_i) \in M_{rcv}(\sigma)\}$

Intuitively, C_{snt} (resp. C_{rvd}) represents the set of all credentials sent (resp. received) and lp_{snt} (resp. lp_{rvd}) represents the last policy sent (resp. received).

Definition 5 (Negotiation State Machine) *A Negotiation State Machine is a tuple*

$$(\Sigma, S, s_0, t)$$

such that

- $S \equiv$ *a set of Negotiation States*
- $s_0 \equiv$ *the empty list (initial state)*

- $\Sigma \equiv$ *a set of Negotiation Messages.*
- $t \equiv$ *a function* $S \times \Sigma \rightarrow S$ *such that if* $S = (\sigma_1, \ldots \sigma_n)$ *then* $t(S, \sigma) = (\sigma_1, \ldots \sigma_n, \sigma_{n+1})$ *(transition function)*

Intuitively a Negotiation State Machine models how an entity evolves during the negotiation by the exchange of messages. Σ contains both sent and received Negotiation Messages and can therefore be partitioned into two subsets Σ_{snd} and Σ_{rcv}.

Definition 6 (Negotiation Model) *A Negotiation Model is a tuple*

$$(C, P, p_0, NSM, ff, ns)$$

where

- $C \equiv$ *a set of credentials*
- $P \equiv$ *a set of Policies*
- $p_0 \equiv$ *a Policy (local Policy)*
- $NSM \equiv$ *a Negotiation State Machine* (Σ, S, s_0, t)
- $ff \equiv$ *a function* $S \rightarrow P$ *(Filtering Function)*
- $ns \equiv$ *an ordered pair* (csf, ta) *where*
  - $csf \equiv$ *a function* $S \rightarrow C$ *(Credential Selection Function)*
  - $ta \equiv$ *a function* $S \rightarrow \{true, false\}$ *(Termination Algorithm)*

  (Negotiation Strategy)

Each occurrence of S is supposed to refer to the same set of Negotiation States.

The intended meaning is as follows

- C represents the set of the credentials local to the Peer
- p_0 represents the Peer's policy protecting the local credentials and allowing access to the local resources
- S represents the set of states in which the Peer can be
- s_0 represents the initial state, i.e. the state in which the Peer is at the beginning of the negotiation
- f represents the process of filtering the Peer's Policy according to the current state
- csf represents the process of selecting the Peer's credentials to send to the other Peer
- ta represent the Peer's decision about whether going on or terminating the current negotiation