Delegation in demexp

BY DAVID MENTRÉ

version: 2004-11-15

This article presents how delegation is handled within demexp. Simultaneously to an informal presentation, we try to give a more formal definition of delegation.

1 Manipulated entities

Demexp is made of a set of questions Q. Each question is marked by zero or more tags belonging to a set T. Tags are unique labels identifying a subject or a domain (e.g. Rennes, Environment, ...). A question q has a set of tags $\tau(q)$.

Definition 1. *Q is the set of questions, T is the set of tags.*

Definition 2. $\forall q \in Q, \tau(q) = \{t | t \text{ is a tag of } q\}$

Note 3. A requirement of the delegation system is to be able to delegate individual questions. This is done by creating a unique tag t_q for each question q and delegating over this tag t_q.

In demexp, we differentiate between *delegates* and *individuals*. Individuals are real people. Delegates are a second entity to each real people to whom individuals or other delegates can delegate a set of tags. A *participant* is either an individual or a delegate.

Definition 4. *I is the set of individuals, D the set of delegates and P the set of Participants. We have:*
$P = I \cup D$
$I \cap D = \varnothing$

2 Delegation

A delegation of a participant p to a delegate d for the tag t is the explicit acknowledgment of the participant p that for all questions that have t in their tags, the delegate d can vote instead of participant p. The participant p can no longer vote on questions having the tag t.

Definition 5. *A delegation from participant p to delegate d for tag t is written:*
$p \xrightarrow{t} d$, $p \in P, d \in D, t \in T$.

Delegation is transitive. In other words, a delegate can delegate in turn a tag to another delegate.

Definition 6. *A chain of delegation $p \rightsquigarrow^t d_n$ is the longest chain such that:*
$p \xrightarrow{t} d_1 \xrightarrow{t} d_2 \xrightarrow{t} \ldots \xrightarrow{t} d_n, t \in T, p \in P, d_1, \ldots, d_n \in D$

A participant cannot delegate to itself because this has no meaning.

Axiom 7. *A participant cannot delegate, directly or indirectly to itself.*
$\forall t \in T, \forall p \in P, p \rightsquigarrow^t p$ *is impossible.*

3 Conflict

When a participant p delegates, a delegation conflict can happen. For example, consider that participant david delegates tag Rennes to delegate A and tag Environment to delegate B. For a question having *both* tags Rennes and Environment, which delegate of A and B should be allow to vote?

Definition 8. *A conflict* $c(p, q, t, t')$ *for participant* p *over question* q *for tags* t *and* t' *is a case where:*
$\exists p \in P, \exists d, d' \in D, \exists q \in Q, \exists t, t' \in \tau(q)$
$p \rightsquigarrow^t d \wedge p \rightsquigarrow^{t'} d'$

To solve a conflict, a participant defines his preference between the two conflicting tags by defining an personal order over tags. In other words, on above example, the participant p tells of which tags Rennes and Environment it is more important than the other one. The delegate corresponding to the more important tag will take precedence over the other delegate and will be able to vote.

Definition 9. *An order* $\succ_p$ *for a participant* p *over tags is a relation such that:*
$\forall t, t' \in T, t \succ_p t' \vee t' \succ_p t$

The ordering relation over tags is transitive. For example, if a participant prefers tag t to t' and t' to t'' then he prefers t to t''.

Axiom 10. *Tag ordering relation* $\succ_p$ *is transitive, i.e.* $\forall t, t', t'' \in T, t \prec_p t' \wedge t' \prec_p t'' \Rightarrow t \prec_p t''$.

A conflict is solved by removing the delegation corresponding to the less-preferred tag.

Axiom 11. *A conflict* $c(p, q, t, t')$ *is solved with tag ordering relation* $\succ_p$ *by removing less-preferred delegation:*
$\exists p \in P, \exists d, d' \in D, \exists q \in Q, \exists t, t' \in \tau(q)$
$p \rightsquigarrow^t d \wedge p \rightsquigarrow^{t'} d' \wedge t \succ_p t' \Rightarrow p \rightsquigarrow^t d \wedge t \succ_p t'$
(and $p \rightsquigarrow^{t'} d'$ *is removed)*

Note 12. When we say $p \rightsquigarrow^{t'} d'$ is removed, we do no say exactly what is removed: the whole delegation chain? The latest delegation? To be defined.

4 Vote weight

When a participant votes on a question q, his vote has an integer weight. This weight is 1 if he is an individual and if the question has not been delegated, 0 otherwise. For a delegate, his weight is equal to the amount of individuals who have delegated to him for a tag of the question. In case nobody has delegated to him, his weight is zero.

Definition 13. *The weight* $w(i, q)$ *of an individual* i *on a question* q *is defined as:*

$$\begin{aligned} \forall i \in I, \forall q \in Q, w(i, q) &= 0 \,\text{if}\, \exists t \in \tau(q) \wedge \exists d \in D \mid i \rightsquigarrow^t d \\ &= 1 \,\text{otherwise} \end{aligned}$$

Definition 14. *The weight* $w(d, q)$ *of a delegate* d *on a question* q *is defined as:*

$$\forall d \in D, \forall q \in Q, w(d, q) = \sum_{\forall t \in \tau(q), \forall i \in I} \begin{matrix} 1 & \text{if}\, i \rightsquigarrow^t d \\ \\ 0 & \text{if}\, \exists t, t' \in \tau(q) | c(i, q, t, t') \end{matrix}$$

One should notice that in above definition, we only consider delegation starting from an *individual* ($i \rightsquigarrow^t d$).

5 Algorithms

To be defined.