



**HAL**  
open science

## Trust in complex actions (ESSLLI 2010)

Julien Bourdon, Guillaume Feuillade, Andreas Herzig, Emiliano Lorini

► **To cite this version:**

Julien Bourdon, Guillaume Feuillade, Andreas Herzig, Emiliano Lorini. Trust in complex actions (ESSLLI 2010). Workshop Logics in Security @ 22rd European Summer School in Logic, Language and Information (ESSLLI 2010), Aug 2010, Copenhagen, Denmark. hal-03470307

**HAL Id: hal-03470307**

**<https://hal.science/hal-03470307v1>**

Submitted on 8 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Trust in complex actions

Julien Bourdon<sup>1</sup>, Guillaume Feuillade<sup>2</sup>, Andreas Herzig<sup>2</sup>, and Emiliano Lorini<sup>2</sup>

1. Kyoto University, Department of Social Informatics, Japan
2. Université de Toulouse, CNRS, IRIT, France

**Abstract.** Current formal models of trust are limited since they only consider an agent’s trust in the atomic action of another agent and therefore do not apply to trust in complex actions where the elements in the complex action are atomic actions of different agents. Our aim is to present a logical formalization of trust in complex actions, and to show that this formalization can be useful for the formal characterization of trust in composite services, where trust in a composed service is defined in a compositional way from trust in the components of that service.

## 1 Introduction

According to Castelfranchi and Falcone (C&F henceforth), the trust relation involves a truster  $i$ , a trustee  $j$ , an action  $a$  that is performed by  $j$  and a goal  $\varphi$  of  $i$  [5, 7]. They defined the predicate **Trust** as a goal together with a particular configuration of beliefs of the trustee. Precisely,  $i$  trusts  $j$  to do  $a$  in order to achieve  $\varphi$  if and only if:

1.  $i$  has the goal that  $\varphi$  and
2.  $i$  believes that:
  - (a)  $j$  is capable to perform  $a$ ,
  - (b)  $j$  is willing to perform  $a$ ,
  - (c)  $j$  has the power to achieve  $\varphi$  by doing  $a$ .

C&F distinguish external from internal conditions in trust assessment:  $j$ ’s capability to perform  $a$  is an external condition, while  $j$ ’s willingness to perform  $a$  is an internal condition (being about the trustee’s mental state). Finally,  $j$ ’s power to achieve  $\varphi$  by doing  $a$  relates internal and external conditions: if  $j$  performs  $a$  then  $\varphi$  will result. Observe that in the power condition the result is conditioned by the execution of  $a$ ; therefore the power condition is independent from the capability condition. In particular,  $j$  may well have the power to achieve  $\varphi$  without being capable to perform  $a$ : for example, right now I have the power to lift a weight of 50kg, but I am not capable to do this because there is no such weight at hand.<sup>1</sup>

C&F did not investigate further how goals, capabilities, willingness and power have to be defined; their definition might therefore be called semi-formal. Recently Herzig,

---

<sup>1</sup> Together, capability to perform  $a$  and power to achieve  $\varphi$  by doing  $a$  amount to having a strategy to achieve  $\varphi$ . Similar modalities were studied in Coalition Logic CL [17], Alternating-time Temporal Logic ATL [1], and STIT theory [2]. However, these logics focus on game-theoretic situations where an agent has the power to achieve  $\varphi$  *whatever the other agents choose to do*. While this latter aspect will not be captured in our analysis here, we have shown in [12, 15] how it could be integrated into our logical framework.

Lorini et al. analysed these predicates in more detail in [13, 14, 16]. First of all they defined the predicate  $\text{Trust}_0$  as follows:

$$\text{Trust}_0(i, j, a, \varphi) \stackrel{\text{def}}{=} \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(i:a) \wedge \text{CInt}(i:a) \wedge \text{Res}(i:a, \varphi))$$

where  $\text{Goal}(i, \varphi)$  corresponds to item 1 and  $\text{CExt}(i:a)$ ,  $\text{CInt}(i:a)$  and  $\text{Res}(i:a, \varphi)$  respectively correspond to items 2a, 2b and 2c in C&F's definition (and  $\text{CExt}$  and  $\text{CInt}$  stand for the external and the internal conditions in trust assessment).<sup>2</sup>

They then defined the predicates  $\text{CExt}(j:a)$ ,  $\text{CInt}(j:a)$  and  $\text{Res}(j:a, \varphi)$  in terms of the concepts of belief, choice, action and time. They draw a distinction between occurrent trust — $i$ 's trust that  $j$  is going to perform  $a$  here and now— and dispositional trust:  $i$  trusts that  $j$  is going to perform  $a$  whenever suitable conditions obtain. The above definition is that of occurrent trust.

Both C&F and Herzig, Lorini et al. only considered trust in the atomic action of another agent and did not consider trust in complex actions where the elements in the complex action are atomic actions of different agents. Our aim in this article is to extend their definition to complex actions, and to show that a definition of trust in complex actions is extremely important in the context of composite services.

In the context of services architecture, some provider agents publish atomic services; however, when a client agent needs to implement a more complex business process, it must chain service calls, according to a specific workflow structure. Automating the service calls is called service composition: given the business process to implement, the control flow has to be computed in order to guarantee that the goal of the service caller is satisfied.

Since services are provided by agents, users may trust some agents for certain actions but not for other actions which they deem critical, usually depending on the nature of the information they have to send to this agent for the service action to perform the action.

In current literature, for example in [18], service selection for composition assumes the existence of a central authority guaranteeing the non-functional properties of the services. In practice, such an authority might not exist, for example in P2P networks [20], or may not itself be trustworthy.

To resolve the aforementioned problem, trust in complex action, supported by the introduction of beliefs and trust in the description of services, could be used. With a model for trust in the services world, one may express composition objectives as dynamic logic formulas with trust component. The service composition problem would then reduce to the satisfaction of such a formula. This method would ensure that the composition is correct and compatible with the beliefs of the user, thus ensuring a trustworthy sequence of service call for achieving the goal of the user.

The rest of the paper is organized as follows. In Section 2 we introduce a modal logic of belief, goal, time, and complex actions. In Section 3 we define the trust predicate for complex actions and study its constituents. In Section 4 we relate trust in complex

<sup>2</sup> They used a 4-ary predicate  $\text{Trust}(i, j, a, \varphi)$  instead of our ternary  $\text{Trust}_0(i, j, a, \varphi)$ . Moreover, their trust definition was in terms of the predicates  $\text{Capable}(j:a)$ ,  $\text{Willing}(j:a)$  and  $\text{Power}(j:a, \varphi)$  instead of our  $\text{CExt}(j:a)$ ,  $\text{CInt}(j:a)$  and  $\text{Res}(j:a, \varphi)$ . We preferred our terms and notations because they better generalize to complex actions.

actions to trust in atomic actions, providing thus a way to construct trust in complex actions. In Section 5 we then discuss how this applies to service composition.

## 2 Background

We recall the logical framework of [13, 14, 16], that we extend towards complex actions.

### 2.1 A logical language with complex actions

Suppose given three countable sets: a set of propositional variables  $Atm$ , a set of agents  $Agt$  and a set of atomic actions  $Act$ . Complex formulas  $\varphi$  and complex actions  $\alpha$  are defined by the following BNF:

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \text{Bel}_i\varphi \mid \text{Ch}_i\varphi \mid \text{Feasible}_\alpha\varphi \mid \text{Happens}_\alpha\varphi \mid \text{F}\varphi \\ \alpha &::= i:a \mid \alpha; \alpha \mid \alpha+\alpha \mid \varphi? \mid \alpha^*\end{aligned}$$

where  $p$  ranges over  $Atm$ ,  $i$  ranges over  $Agt$ , and  $a$  ranges over  $Act$ .  $\text{Bel}_i\varphi$  reads “ $i$  believes that  $\varphi$ ”;  $\text{Ch}_i\varphi$  reads “ $i$  chooses that  $\varphi$ ”;  $\text{Feasible}_\alpha\varphi$  reads “there is a possible execution of  $\alpha$  after which  $\varphi$  is true”;  $\text{Happens}_\alpha\varphi$  reads “ $\alpha$  happens, and  $\varphi$  is true afterwards”; and  $\text{F}\varphi$  reads “ $\varphi$  will eventually be true”. Both  $\text{Feasible}_\alpha$  and  $\text{Happens}_\alpha$  are modal operators of the possible kind, and could be written  $\langle\alpha\rangle$  as in dynamic logic. Note that they are different:  $\text{Feasible}_\alpha\top$  means that  $\alpha$  is executable, while  $\text{Happens}_\alpha\top$  means that  $\alpha$  is executed.

Operators  $\text{Ch}_i$  are used to denote an agent’s current chosen goals, that is, the goals that the agent has decided to pursue here and now. We do not consider how an agent’s chosen goals originate through deliberation from more primitive motivational attitudes, called desires, and from moral attitudes, such as ideals and imperatives. Since the chosen goals of an agent result from the its deliberation, they must satisfy two fundamental rationality principles: chosen goals have to be consistent (i.e., a rational agent cannot decide to pursue an inconsistent state of affairs); chosen goals have to be compatible with the agent’s beliefs (i.e., a rational agent cannot decide to pursue something that it believes to be impossible). These two principles will be formally expressed in Section 2.2.

*Remark 1.* In [13, 14, 16]  $\text{Happens}_{i:a}$  was noted  $\text{Does}_{i:a}$  and read “ $a$  is going to be performed by  $i$ ”. We preferred  $\text{Happens}$  in order to allow for complex actions such as  $i:a; j:b$  that are performed by more than one agent.

The atomic action  $i:a$  reads “ $i$  performs  $a$ ”; the complex action  $\alpha_1; \alpha_2$  reads “do  $\alpha_1$  and then  $\alpha_2$ ”;  $\alpha_1+\alpha_2$  reads “choose nondeterministically between  $\alpha_1$  and  $\alpha_2$ ”, where the choice is understood to be up to the environment (i.e., the other agents and nature), and not up to the agents performing  $\alpha_1$  and  $\alpha_2$ ;  $\varphi?$  reads “if  $\varphi$  is true then continue, else fail”; and finally,  $\alpha^*$  reads “do  $\alpha$  an arbitrary number of times”.

We define  $\text{After}_\alpha\varphi$  to be an abbreviation of  $\neg\text{Feasible}_\alpha\neg\varphi$ , which therefore has to be read “if the execution of  $\alpha$  is possible then  $\varphi$  holds afterwards”. Moreover, the following standard program constructions are defined as follows:

$$\begin{aligned}
\text{skip} &\stackrel{\text{def}}{=} \top? \\
\text{fail} &\stackrel{\text{def}}{=} \perp? \\
\text{if } \varphi \text{ then } \alpha_1 \text{ else } \alpha_2 &\stackrel{\text{def}}{=} (\varphi?; \alpha_1) + (\neg\varphi?; \alpha_2) \\
\text{while } \varphi \text{ do } \alpha &\stackrel{\text{def}}{=} (\varphi?; \alpha)^*; \neg\varphi?
\end{aligned}$$

In our application the actions seem never to be joined actions (which here would be something like translating a text together). For that reason we define parallel composition as interleaving, i.e.  $\alpha \parallel \beta \stackrel{\text{def}}{=} (\alpha; \beta) + (\beta; \alpha)$ . In this way we can avoid introducing  $\parallel$  as a primitive.

## 2.2 Semantics

We take over the semantics of [14] and extend it to complex actions, whose semantics we take over from Propositional Dynamic Logic PDL. We call the resulting logic  $\mathcal{L}$ . The semantics of  $\mathcal{L}$  is in terms of a class of frames that has to satisfy several constraints.

**Frames** A *frame* is a tuple  $M = \langle W, A, B, C, D \rangle$  that is defined as follows.

- $W$  is a nonempty set of possible worlds or states.
- $A : \text{Agt} \times \text{Act} \rightarrow 2^{W \times W}$  maps every agent  $i$  and action  $a$  to a relation  $A_{i,a}$  between possible worlds in  $W$ .
- $B : \text{Agt} \rightarrow 2^{W \times W}$  maps every agent  $i$  to a serial, transitive and Euclidean<sup>3</sup> relation  $B_i$  between possible worlds in  $W$ .
- $C : \text{Agt} \rightarrow 2^{W \times W}$  maps every agent  $i$  to a serial relation  $C_i$  between possible worlds in  $W$ .
- $D : \text{Agt} \times \text{Act} \rightarrow 2^{W \times W}$  maps every agent  $i$  and action  $a$  to a deterministic relation (alias a partial function)  $D_{i,a}$  between possible worlds in  $W$ .<sup>4</sup>

It is convenient to view relations on  $W$  as functions from  $W$  to  $2^W$ ; therefore we write  $D_{i,a}(w)$  for the set  $\{w' \mid (w, w') \in D_{i,a}\}$ , etc.

When  $w' \in A_{i,a}(w)$  then *if* at  $w$  agent  $i$  performs  $a$  then this *might* result in  $w'$ .  $B_i(w)$  is the set of worlds that are compatible with agent  $i$ 's beliefs at  $w$ ; the conditions of seriality, transitivity and Euclideanity are those of the standard logic of belief KD45.  $C_i(w)$  is the set of worlds that are compatible with agent  $i$ 's choices at  $w$ ; seriality corresponds to consistency of choices, which is the only condition that is generally imposed on choices.  $D_{i,a}(w)$  is the set of worlds  $w'$  that can be reached from  $w$  through the occurrence of agent  $i$ 's action  $a$ . If  $(w, w') \in D_{i,a}$  then  $w'$  is the unique actual *successor* world of  $w$ , that will be reached from  $w$  through the occurrence of agent  $i$ 's action  $a$  at  $w$ : at  $w$  agent  $i$  performs an action  $a$ , resulting in the next state  $w'$ . (We might also say that  $D_{i,a}$  is a partial function.) If  $D_{i,a}(w) \neq \emptyset$  then we say that  $D_{i,a}$  is defined at  $w$ .

<sup>3</sup> A relation  $B_i$  on  $W$  is Euclidean if and only if, if  $(w, w') \in B_i$  and  $(w, w'') \in B_i$  then  $(w', w'') \in B_i$ .

<sup>4</sup> A relation  $D_{i,a}$  is deterministic iff, if  $(w, w') \in D_{i,a}$  and  $(w, w'') \in D_{i,a}$  then  $w' = w''$ .

**Constraints on frames** Frames will have to satisfy some constraints in order to be legal  $\mathcal{L}$ -frames. For every  $i, j \in \text{Agt}$ ,  $\alpha, \beta \in \text{Act}$  and  $w \in W$  we suppose:

**C1** if  $D_{i:a}$  and  $D_{j:b}$  are defined at  $w$  then  $D_{i:a}(w) = D_{j:b}(w)$ .

Constraint **C1** says that if  $w'$  is the *next* world of  $w$  which is reached from  $w$  through the occurrence of agent  $i$ 's action  $\alpha$  and  $w''$  is also the *next* world of  $w$  which is reached from  $w$  through the occurrence of agent  $j$ 's action  $\beta$ , then  $w'$  and  $w''$  denote the same world. Indeed, we suppose that one agent acts at a time, and that every world can only have one *next* world. Note that **C1** implies determinism of every  $D_{i:a}$  (so we might have omitted that from the above constraints on  $D$ ).

Therefore, when  $w' \in A_{i:a}(w)$  but  $D_{i:a}(w) = \emptyset$  then at  $w$  agent  $i$  does not perform  $\alpha$ , but if it did so it might have produced another outcome world  $w'$ .

Moreover, for every  $i \in \text{Agt}$ ,  $\alpha \in \text{Act}$  we suppose:

**C2**  $D_{i:a} \subseteq A_{i:a}$ .

The constraint **C2** says that if  $w'$  is the *next* world of  $w$  which is reached from  $w$  through the occurrence of agent  $i$ 's action  $\alpha$ , then  $w'$  must be a world which is *reachable* from  $w$  through the occurrence of agent  $i$ 's action  $\alpha$ .

The next constraint **C3** links the agents' choices with what they do: if  $a$  is executable and  $i$  chooses to do  $a$  then  $a$  is going to happen.

**C3** if  $A_{i:a}$  is defined at  $w$  and  $D_{i:a}$  is defined at  $w'$  for all  $w' \in C_i(w)$  then  $D_{i:a}$  is defined at  $w$ .

The following semantic constraint **C4** is also about the relationship between an agent  $i$ 's choices (i.e., chosen worlds) and the actions performed by  $i$ . For every  $i \in \text{Agt}$ ,  $\alpha \in \text{Act}$  and  $w \in W$ , we suppose that:

**C4** if  $w' \in C_i(w)$  and  $D_{i:a}$  is defined at  $w$ , then  $D_{i:a}$  is defined at  $w'$ .

In other words, if it is not the case that  $i$  performs  $a$  in all of  $i$ 's chosen worlds then  $i$  is not going to perform  $a$ .

The next constraint relates worlds that are compatible with agent  $i$ 's beliefs and worlds that are compatible with  $i$ 's chosen goals: as motivated in the beginning of Section 2.1, they should not be disjoint. For every  $i \in \text{Agt}$  and  $w \in W$ :

**C5**  $C_i(w) \cap B_i(w) \neq \emptyset$ .

The next constraint on  $\mathcal{L}$ -frames is one of introspection w.r.t. choices. For every  $i \in \text{Agt}$  and  $w \in W$ :

**C6** if  $w' \in B_i(w)$  then  $C_i(w) = C_i(w')$ .

The next two constraints on  $\mathcal{L}$ -frames are what is called 'no learning' and 'no forgetting' for beliefs in the literature [6]. For every  $i, j \in \text{Agt}$ ,  $a \in \text{Act}$  and  $w \in W$ :

**C7** if  $(w, v) \in A_{j:a} \circ B_i$  and there is  $u$  such that  $(w, u) \in B_i \circ A_{j:a}$  then  $(w, v) \in B_i \circ A_{j:a}$

**C8** if  $(w, v) \in B_i \circ A_{j:a}$  and there is  $u$  such that  $(w, u) \in A_{j:a}$  then  $(w, v) \in A_{j:a} \circ B_i$ ,

where  $\circ$  is the standard composition operator between two binary relations. Thus, we suppose that events are always uninformative, in the sense: *i should not forget* anything about the particular effects of *j*'s action *a* that starts at a given world *w*, and *i should not learn* anything new due to the occurrence of *j*'s action *a* that starts at a given world *w* (except the occurrence of that very action). In other words, what an agent *i* believes at a world *v* after the occurrence of *j*'s action *a*, only depends on what *i* believed at the previous world *w* and on the action which has occurred and which was responsible for the transition from *w* to *v*. Note that the ‘no forgetting’ and ‘no learning’ constraints rely on an additional assumption that actions are public: it is supposed that *j*'s action *a* occurs if and only if every agent is informed of this fact.

We have similar principles of no learning and no forgetting for the relations  $D_{i:a}$ . For every  $i, j \in \text{Agt}$ ,  $a \in \text{Act}$  and  $w \in W$ :

- C9** if  $(w, v) \in D_{j:a} \circ B_i$  and there is  $u$  such that  $(w, u) \in B_i \circ A_{j:a}$  then  $(w, v) \in B_i \circ D_{j:a}$ ;  
**C10** if  $(w, v) \in B_i \circ D_{j:a}$  and there is  $u$  such that  $(w, u) \in D_{j:a}$  then  $(w, v) \in D_{j:a} \circ B_i$ .

**Models and truth conditions** A *model* is a tuple  $M = \langle W, A, B, C, D, V \rangle$  where the tuple  $\langle W, A, B, C, D \rangle$  is a frame and  $V : \text{Atm} \rightarrow 2^W$  is a valuation.

Formulas and events are interpreted according to the following clauses.

$$\begin{aligned}
R_{\alpha;\beta} &= R_\alpha \circ R_\beta \\
R_{\alpha+\beta} &= R_\alpha \cup R_\beta \\
R_{\varphi?} &= \{ \langle v, v \rangle \mid v \in W \text{ and } M, v \models \varphi \} \\
R_{\alpha^*} &= (R_\alpha)^* \\
M, w \models p &\text{ iff } w \in V(p) \\
M, w \models \text{Bel}_i \varphi &\text{ iff } M, w' \models \varphi \text{ for every } w' \in B_i(w) \\
M, w \models \text{Ch}_i \varphi &\text{ iff } M, w' \models \varphi \text{ for every } w' \in C_i(w) \\
M, w \models \text{Feasible}_\alpha \varphi &\text{ iff } M, w' \models \varphi \text{ for some } w' \in A_\alpha(w) \\
M, w \models \text{Happens}_\alpha \varphi &\text{ iff } M, w' \models \varphi \text{ for some } w' \in D_\alpha(w) \\
M, w \models \text{F}\varphi &\text{ iff } M, w' \models \varphi \text{ for some } w' \text{ such that } w(\bigcup_{a \in \text{Act}} D_a)^* w'
\end{aligned}$$

The clauses for the Boolean operators are as usual. The last clause is based on the hypothesis that time flow is determined by the actions that are performed (where the  $D_a$  and  $\bigcup_{a \in \text{Act}} D_a$  are understood as relations).  $\bigcup_{a \in \text{Act}} D_a(w)$  is the set of worlds  $w'$  that are in the future of  $w$ :  $w'$  can be attained from  $w$  by some  $D$ -chain, i.e. by some sequence of actions.

### 2.3 Some useful validities

We now state some validities of our logic that will be useful later.<sup>5</sup>

**Proposition 1.** *The following formulas are valid:*

<sup>5</sup> We do not give a completeness result: there is such a result (albeit for a simpler language) in [14], which should be extended in order to account for complex actions; in particular the Kleene star “\*” requires a fixpoint axiom and a least fixpoint axiom, which makes that the completeness proof is not straightforward.

1.  $\text{After}_{\varphi?}\psi \leftrightarrow (\varphi \rightarrow \psi)$
2.  $\text{Happens}_{\varphi?}\psi \leftrightarrow (\varphi \wedge \psi)$
3.  $\text{Happens}_{\alpha\beta}\varphi \leftrightarrow (\text{Happens}_{\alpha}\top \wedge \neg\text{Happens}_{\alpha}\neg\text{Happens}_{\beta}\varphi)$
4.  $\text{Happens}_{\alpha+\beta}\varphi \leftrightarrow (\text{Happens}_{\alpha}\varphi \vee \text{Happens}_{\beta}\varphi)$
5.  $\text{Happens}_{\alpha^*}\varphi \leftrightarrow (\varphi \wedge \text{Happens}_{\alpha}\text{Happens}_{\alpha^*}\varphi)$
6.  $(\text{F}\neg\varphi \wedge \text{After}_{\alpha^*}(\varphi \rightarrow \text{Happens}_{\alpha}\top)) \rightarrow \text{Happens}_{\text{while } \varphi \text{ do } \alpha}\top$
7.  $\text{Happens}_{\alpha}\varphi \rightarrow \text{Feasible}_{\alpha}\varphi$
8.  $\text{Feasible}_{\alpha}\varphi \rightarrow \text{F}\varphi$
9.  $(\text{Happens}_{\alpha}\varphi \wedge \text{After}_{\alpha}\psi) \rightarrow \text{Happens}_{\alpha}(\varphi \wedge \psi)$
10.  $\neg(\text{Ch}_i\varphi \wedge \text{Bel}_i\varphi)$
11.  $(\text{Feasible}_{i:a}\varphi \wedge \text{Ch}_i\text{Happens}_{i:a}\top) \rightarrow \text{Happens}_{i:a}\top$
12.  $(\neg\text{Bel}_i\neg\text{Feasible}_{\alpha}\top \wedge \text{Bel}_i\text{After}_{\alpha}\varphi) \rightarrow \text{After}_{\alpha}\text{Bel}_i\varphi$
13.  $(\text{Feasible}_{\alpha}\top \wedge \text{After}_{\alpha}\text{Bel}_i\varphi) \rightarrow \text{Bel}_i\text{After}_{\alpha}\varphi$
14.  $(\neg\text{Bel}_i\neg\text{Happens}_{\alpha}\top \wedge \text{Bel}_i\neg\text{Happens}_{\alpha}\neg\varphi) \rightarrow \neg\text{Happens}_{\alpha}\neg\text{Bel}_i\varphi$
15.  $(\text{Happens}_{\alpha}\top \wedge \neg\text{Happens}_{\alpha}\neg\text{Bel}_i\varphi) \rightarrow \text{Bel}_i\neg\text{Happens}_{\alpha}\neg\varphi$

Formula 11 is a principle of intentional action **IntAct**. The last four are principles of no forgetting (**NF**, alias perfect recall) and no learning (**NL**, alias no miracles) for beliefs. Similar principles have been studied in [8, 19, 11].

### 3 Trust about complex actions

We now generalize the definition of (occurrent) trust about atomic actions of [13, 14, 16] to trust about complex actions and study its constituents. Among all possible complex actions we here only consider deterministic actions [9]: actions built with “skip”, “fail”, “;”, “if  $\varphi$  then  $\alpha_1$  else  $\alpha_2$ ”, and “while  $\varphi$  do  $\alpha$ ”. Their BNF is:

$$\alpha ::= i:a \mid \text{skip} \mid \text{fail} \mid \alpha;\alpha \mid \text{if } \varphi \text{ then } \alpha \text{ else } \alpha \mid \varphi? \mid \text{while } \varphi \text{ do } \alpha$$

Tests  $\varphi?$  can be defined as **if  $\varphi$  then skip else fail**. In our analysis of trust in complex actions we do not consider the other program operators of PDL, viz. nondeterministic composition and iteration.

Let us first recall the definition of the original trust predicate in [13, 14, 16]. There, the goal condition  $\text{Goal}(i, \varphi)$  was defined as  $\text{Ch}_i\text{F}\varphi$ , i.e. as  $i$ 's choice of futures where  $\varphi$  holds. The external condition  $\text{CExt}(j:a)$  was defined as  $\text{Feasible}_{j:a}\top$  ( $j:a$  is executable), and the internal condition  $\text{CInt}(j:a)$  as  $\text{Ch}_j\text{Happens}_{j:a}\top$  ( $j$  chooses that  $j:a$  is going to occur). Finally, the power condition  $\text{Res}(j:a, \varphi)$  was defined as  $\text{After}_{j:a}\varphi$  ( $\varphi$  will hold immediately after every possible performance of  $j:a$ ).

It turns out that our move from trust in atomic actions to trust in complex actions requires some adjustments.

#### 3.1 Definition of trust

First of all, here is our official definition of trust in a complex action:

$$\text{Trust}(i, \alpha, \varphi) \stackrel{\text{def}}{=} \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha) \wedge \text{Res}(\alpha, \varphi))$$



where  $i$  is an agent,  $\alpha$  is a deterministic action, and  $\varphi$  is a formula. As before, **CExt** and **CInt** stand for the external and the internal conditions in trust assessment; they will be defined in the sequel. We have thus simply replaced the atomic actions in our definition of Section 1 by complex actions.

Observe that trust in atomic actions involved a single trustee  $j$ . Here we have to account for trust in complex actions that may be performed by several agents; we therefore consider trust in a group of agents.

Note also that before, the trustee  $j$ —which here would be a set of agents  $J$ —appeared explicitly in the definition of the predicate **Trust**. However, one may consider that  $J$  is implicitly already there: it is the set of agents occurring in  $\alpha$ . Therefore the agent argument need not appear as a separate argument in the definition.

It remains to explain the predicates on the right hand side of the definition of trust.

### 3.2 Defining the ingredients of trust

We now reduce the predicates on the right hand side of the definition of trust.

**Goal** The definition of the **Goal** predicate transfers straightforwardly because no action occurs in it:

$$\text{Goal}(i, \varphi) \stackrel{\text{def}}{=} \text{Ch}_i \text{F}\varphi$$

So it remains to define **CExt**, **CInt** and **Res**.

**Result** The original power condition  $\text{Bel}_i \text{After}_{j:a} \varphi$  stipulated that  $i$  believes  $\varphi$  immediately results from  $j$ 's performance of atomic action  $a$ . However, consider  $i$ 's trust in  $j_1$  and  $j_2$  to perform the sequence of actions  $j_1:a_1; j_2:a_2$  in order to achieve  $i$ 's goal  $\varphi$ . With respect to which goal should  $i$  trust  $j_1$ ? The truster  $i$  typically does not bother about the direct effect of  $j_1$ 's action  $a_1$  and is only interested in the overall effect  $\varphi$  of the complex action  $j_1:a_1; j_2:a_2$ . In other words, we have to account for the case where  $\varphi$  is not achieved immediately, but only at some time point in the future. We therefore redefine

$$\text{Res}(\alpha, \varphi) \stackrel{\text{def}}{=} \text{After}_\alpha \text{F}\varphi$$

Under the other definitions to come, the original  $\text{Trust}_0(i, j:a, \varphi)$  will be equivalent to our  $\text{Trust}(i, j:a, \text{F}\varphi)$ .

**External and internal condition** Up to now, all our definitions were directly in terms of well-defined formulas of our logic. Things are not as simple for the external condition **CExt** and for the internal condition **CInt**.

In [13, 14, 16], using axiom **IntAct** it was proved that

$$(\text{CExt}(i:a) \wedge \text{CInt}(i:a)) \rightarrow \text{Happens}(i:a)$$

is valid. That is, if both the external condition and the internal condition for the execution of action  $a$  by agent  $i$  obtain—i.e.,  $i$  is capable to perform action  $a$  and is willing (has the intention) to perform  $a$ —then  $i$  performs  $a$ . We would like to keep this principle of intentional action, and therefore need a definition of the **CExt** and **CInt** predicates

validating

$$(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \rightarrow \text{Happens}(\alpha)$$

In particular, we will have to include a condition guaranteeing that while-loops are exited (because  $\text{Happens}_{\text{while } \psi \text{ do } \alpha} \top$  implies that  $F\neg\psi$ ).

As to the external condition,  $\text{CExt}(\alpha)$  means that the complex action  $\alpha$  is executable *whatever the other agents and nature choose to do*. This means that the preconditions of  $\alpha$  must obtain at every step of every execution of  $\alpha$ . It follows that while  $\text{CExt}(\alpha)$  implies  $\text{Feasible}_\alpha \top$ , it should not be equivalent to it. For example, the complex action  $(i:a+i:a'); i:b$  cannot be said to be executable (in the above sense) when just  $\text{Feasible}_{i:a+i:a'; i:b} \top$  holds. Indeed, a situation where  $\text{Feasible}_{i:a'} \text{After}_{i:b} \top$  is compatible with the latter formula, and if nature chooses  $i:a'$  when executing the nondeterministic  $i:a+i:a'$  then it cannot be said that  $\text{Feasible}_{i:a+i:a'; i:b} \top$  is executable.

Given these considerations we recursively define  $\text{CExt}(\alpha)$  as follows:

$$\begin{aligned} \text{CExt}(i:a) &\stackrel{\text{def}}{=} \text{Feasible}_{i:a} \top \\ \text{CExt}(\text{skip}) &\stackrel{\text{def}}{=} \top \\ \text{CExt}(\text{fail}) &\stackrel{\text{def}}{=} \perp \\ \text{CExt}(\alpha;\beta) &\stackrel{\text{def}}{=} \text{CExt}(\alpha) \wedge \text{After}_\alpha \text{CExt}(\beta) \\ \text{CExt}(\text{if } \varphi \text{ then } \alpha_1 \text{ else } \alpha_2) &\stackrel{\text{def}}{=} (\varphi \wedge \text{CExt}(\alpha_1)) \vee (\neg\varphi \wedge \text{CExt}(\alpha_2)) \\ \text{CExt}(\text{while } \psi \text{ do } \alpha) &\stackrel{\text{def}}{=} F\neg\psi \wedge \text{After}_{(\psi?;\alpha)^*;\psi?} \text{CExt}(\alpha) \end{aligned}$$

It is the clause for “;” that makes that  $\text{CExt}(\alpha)$  stronger than  $\text{Feasible}_\alpha \top$ .

As to the (internal) willingness condition, it is tempting to define  $\text{CInt}(\alpha)$  as

$$\bigwedge_{j \in \text{Agt}(\alpha)} \text{Ch}_j \text{Happens}_\alpha \top,$$

where  $\text{Agt}(\alpha)$  is the set of agent names occurring in  $\alpha$ : every agent involved in the complex action  $\alpha$  chooses that  $\alpha$  happens. However, this would be too strong. Indeed, consider the scenario where  $j_1:a_1$  is  $j_1$ 's action of requesting  $j_2$  to do  $a_2$ , and where  $j_2$  initially prefers not to be asked by  $j_1$ , i.e.  $\text{Ch}_{j_2} \neg \text{Happens}_{j_1:a_1} \perp$ , but intends to perform  $j_2:a_2$  after  $j_1$ 's request. In symbols, we have a situation where  $\text{Happens}_{j_1:a_1;j_2:a_2} \top$  and  $\neg \text{Ch}_{j_2} \text{Happens}_{j_1:a_1;j_2:a_2} \top$  is true.

Such considerations lead to the following recursive definition of the predicate  $\text{CInt}$ .

$$\begin{aligned} \text{CInt}(i:a) &\stackrel{\text{def}}{=} \text{Ch}_i \text{Happens}_{i:a} \top \\ \text{CInt}(\text{fail}) &\stackrel{\text{def}}{=} \top \\ \text{CInt}(\text{skip}) &\stackrel{\text{def}}{=} \top \\ \text{CInt}(\alpha;\beta) &\stackrel{\text{def}}{=} \text{CInt}(\alpha) \wedge \text{After}_\alpha \text{CInt}(\beta) \\ \text{CInt}(\text{if } \varphi \text{ then } \alpha_1 \text{ else } \alpha_2) &\stackrel{\text{def}}{=} (\varphi \wedge \text{CInt}(\alpha_1)) \vee \\ &\quad (\neg\varphi \wedge \text{CInt}(\alpha_2)) \\ \text{CInt}(\text{while } \psi \text{ do } \alpha) &\stackrel{\text{def}}{=} \text{After}_{(\psi?;\alpha)^*;\psi?} \text{CInt}(\alpha) \end{aligned}$$

### 3.3 A principle of intentional action for complex actions

We are now going to relate the predicates CExt and CInt with the modal operator Happens. We prove that when  $\alpha$  is complex then one half of the axiom **IntAct** remains valid.

**Proposition 2.** *The formula  $\text{CExt}(\alpha) \rightarrow \text{Feasible}_\alpha \top$  is valid.*

*Proof.* The proof is by induction on the structure of  $\alpha$ .

As mentioned above, the other direction  $\text{Feasible}_\alpha \top \rightarrow \text{CExt}(\alpha)$  is guaranteed to be valid only when  $\alpha$  is atomic.

**Proposition 3.** *The formula  $(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \rightarrow \text{Happens}_\alpha \top$  is valid.*

*Proof.* We use induction on the structure of  $\alpha$ . The base cases are ensured by the axiom **IntAct** and by Proposition 1. For the induction step we have:

$$\begin{aligned} & \text{CExt}(\alpha; \beta) \wedge \text{CInt}(\alpha; \beta) \\ & \leftrightarrow \text{CExt}(\alpha) \wedge \text{After}_\alpha \text{CExt}(\beta) \wedge \text{CInt}(\alpha) \wedge \text{After}_\alpha \text{CInt}(\beta) \\ & \rightarrow \text{Happens}_\alpha \top \wedge \text{After}_\alpha \text{Happens}_\beta \top \quad (\text{by I.H.}) \\ & \rightarrow \text{Happens}_{\alpha; \beta} \top \quad (\text{by Prop. 1}) \end{aligned}$$

$$\begin{aligned} & \text{CExt}(\text{if } \psi \text{ then } \alpha \text{ else } \beta) \wedge \text{CInt}(\text{if } \psi \text{ then } \alpha \text{ else } \beta) \\ & \leftrightarrow (\psi \rightarrow (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge (\neg\psi \rightarrow (\text{CExt}(\beta) \wedge \text{CInt}(\beta))) \\ & \rightarrow (\psi \rightarrow \text{Happens}_\alpha \top) \wedge (\neg\psi \rightarrow \text{Happens}_\beta \top) \quad (\text{by I.H.}) \\ & \rightarrow \text{Happens}_{\text{if } \psi \text{ then } \alpha \text{ else } \beta} \top \quad (\text{by Prop. 1}) \end{aligned}$$

$$\begin{aligned} & \text{CExt}(\text{while } \psi \text{ do } \alpha) \wedge \text{CInt}(\text{while } \psi \text{ do } \alpha) \\ & \leftrightarrow \text{F}\neg\psi \wedge \text{After}_{(\psi?; \alpha)^*; \psi?} (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \\ & \rightarrow \text{F}\neg\psi \wedge \text{After}_{(\psi?; \alpha)^*} (\psi \rightarrow \text{Happens}_\alpha \top) \quad (\text{by I.H.}) \\ & \rightarrow \text{F}\neg\psi \wedge \text{After}_{(\psi?; \alpha)^*} (\psi \rightarrow \text{Happens}_{\psi?; \alpha} \top) \\ & \rightarrow \text{Happens}_{\text{while } \psi \text{ do } \alpha} \top \quad (\text{by Prop. 1}) \end{aligned}$$

As said above, the other direction of Proposition 3

$$\text{Happens}_{i_1; a_1; i_2; a_2} \top \rightarrow (\text{CExt}(i_1; a_1; i_2; a_2) \wedge \text{CInt}(i_1; a_1; i_2; a_2))$$

is invalid because  $i_1$ 's performance of  $a_1$  may cause  $i_2$ 's performance of  $a_2$ .

We finally observe that when the trustor's goal is  $\top$  then trust in  $\alpha$  amounts to the conjunction of external and internal condition.

**Proposition 4.** *The formula  $\text{Trust}(i, \alpha, \top) \leftrightarrow \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha))$  is valid.*

## 4 Properties of trust

In this section we state the properties of trust in complex actions, alias workflow constructs.

First of all and as announced in Section 3.2 we observe that our and the original definition coincide for atomic actions, except that we have relaxed the result condition:

for us it suffices that the result  $\varphi$  obtains at some point in the future, and not immediately after the action. We therefore have  $\text{Trust}(i, j:a, \varphi) \leftrightarrow \text{Trust}_0(i, j:a, F\varphi)$ .

For complex actions we are going to have reductions in terms of equivalences for the cases of `skip`, `fail`, if-then-else conditionals and while loops. For trust in sequential compositions we only give a sufficient condition. We only give some of the proofs.

#### 4.1 Atomic actions

**Theorem 1.** *The formulas  $\text{Trust}(i, \text{fail}, \varphi) \leftrightarrow \perp$  and  $\text{Trust}(i, \text{skip}, \varphi) \leftrightarrow (\text{Goal}(i, \varphi) \wedge \text{Bel}_i F\varphi)$  are valid.*

#### 4.2 Sequential composition

Our first theorem allows to construct trust in a sequence  $\alpha; \beta$  from trust in  $\alpha$  and trust in  $\beta$ .

**Theorem 2.** *The formula  $(\text{Trust}(i, \alpha, \varphi) \wedge \text{Bel}_i \text{After}_\alpha \text{Trust}(i, \beta, \varphi)) \rightarrow \text{Trust}(i, (\alpha; \beta), \varphi)$  is valid.*

*Proof.* We have:

$$\begin{aligned}
& \text{Trust}(i, \alpha, \varphi) \wedge \text{Bel}_i \text{After}_\alpha \text{Trust}(i, \beta, \varphi) \\
\rightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \\
& \text{Bel}_i \text{After}_\alpha \text{Bel}_i (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta F\varphi) \\
\rightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \\
& \text{Bel}_i (\text{After}_\alpha \perp \vee \text{Bel}_i \text{After}_\alpha (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta F\varphi)) \\
& \hspace{10em} \text{(by NL of Prop. 1)} \\
\rightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \\
& \text{Bel}_i \text{Bel}_i \text{After}_\alpha (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta F\varphi) \\
\rightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \text{Bel}_i \text{After}_\alpha (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta F\varphi) \\
\leftrightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha) \wedge \text{After}_\alpha \text{CExt}(\beta)) \wedge \\
& \text{Bel}_i (\text{CInt}(\alpha) \wedge \text{After}_\alpha \text{CInt}(\beta)) \wedge \text{Bel}_i \text{After}_\alpha \text{After}_\beta F\varphi \\
\leftrightarrow & \text{Goal}(i, \varphi) \wedge \text{Bel}_i (\text{CExt}(\alpha; \beta) \wedge \text{CInt}(\alpha; \beta) \wedge \text{After}_{\alpha; \beta} F\varphi) \\
= & \text{Trust}(i, (\alpha; \beta), \varphi)
\end{aligned}$$

The next two theorems are about the consequences of trust in a sequence of actions.

**Theorem 3.** *The formula  $\text{Trust}(i, (\alpha; \beta), \varphi) \rightarrow \text{Trust}(i, \alpha, \varphi)$  is valid.*

*Proof.* We have:

$$\begin{aligned}
& \text{Trust}(i, (\alpha; \beta), \varphi) \\
& \leftrightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{After}_\alpha \text{CExt}(\beta)) \wedge \\
& \quad \text{Bel}_i(\text{CInt}(\alpha) \wedge \text{After}_\alpha \text{CInt}(\beta)) \wedge \text{Bel}_i \text{After}_{\alpha; \beta} \text{F}\varphi \\
& \leftrightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \\
& \quad \text{Bel}_i \text{After}_\alpha (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta \text{F}\varphi) \\
& \rightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \text{Bel}_i \text{After}_\alpha (\text{Happens}_\beta \top \wedge \text{After}_\beta \text{F}\varphi) \\
& \quad \text{(by Prop. 3)} \\
& \rightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \text{Bel}_i \text{After}_\alpha \text{Happens}_\beta \text{F}\varphi \quad \text{(by Prop. 1)} \\
& \leftrightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i(\text{CExt}(\alpha) \wedge \text{CInt}(\alpha)) \wedge \text{Bel}_i \text{After}_\alpha \text{F}\varphi \quad \text{(by Prop. 1)} \\
& \leftrightarrow \text{Trust}(i, \alpha, \varphi)
\end{aligned}$$

Our last theorem says that trust persists under the condition that the goal persists.

**Theorem 4.** *The formula*

$$\text{Trust}(i, (\alpha; \beta), \varphi) \rightarrow \text{After}_\alpha (\neg \text{Goal}(i, \varphi) \vee \text{Trust}(i, \beta, \varphi))$$

is valid.

*Proof.* First, observe that

$$\text{Trust}(i, (\alpha; \beta), \varphi) \rightarrow \text{Bel}_i \text{Feasible}_\alpha \top \quad (*)$$

is valid by Proposition 3 and Proposition 1. Now:

$$\begin{aligned}
& \text{Trust}(i, (\alpha; \beta), \varphi) \\
& \rightarrow \text{Bel}_i \text{After}_\alpha (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta \text{F}\varphi) \\
& \rightarrow \text{Bel}_i \text{After}_\alpha \perp \vee \text{After}_\alpha \text{Bel}_i (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta \text{F}\varphi) \quad \text{(by NF of Prop. 1)} \\
& \rightarrow \text{After}_\alpha \text{Bel}_i (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta \text{F}\varphi) \quad \text{by } (*)
\end{aligned}$$

### 4.3 If-then-else

**Theorem 5.** *The formula*

$$\text{Trust}(i, \text{if } \psi \text{ then } \alpha \text{ else } \beta, \varphi) \leftrightarrow \text{Bel}_i ( (\psi \rightarrow \text{Trust}(i, \alpha, \varphi)) \wedge (\neg \psi \rightarrow \text{Trust}(i, \beta, \varphi)) )$$

is valid.

*Proof.* We have:

$$\begin{aligned}
& \text{Trust}(i, \text{if } \psi \text{ then } \alpha \text{ else } \beta, \varphi) \\
& \leftrightarrow \text{Goal}(i, \varphi) \wedge \\
& \quad \text{Bel}_i ((\psi \rightarrow \text{CExt}(\alpha)) \wedge (\neg \psi \rightarrow \text{CExt}(\beta))) \wedge \\
& \quad \text{Bel}_i ((\psi \rightarrow \text{CInt}(\alpha)) \wedge (\neg \psi \rightarrow \text{CInt}(\beta))) \wedge \\
& \quad \text{Bel}_i ((\psi \rightarrow \text{After}_\alpha \text{F}\varphi) \wedge (\neg \psi \rightarrow \text{After}_\beta \text{F}\varphi)) \\
& \leftrightarrow \text{Bel}_i \text{Goal}(i, \varphi) \wedge \\
& \quad \text{Bel}_i (\psi \rightarrow (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha) \wedge \text{After}_\alpha \text{F}\varphi)) \wedge \\
& \quad \text{Bel}_i (\neg \psi \rightarrow (\text{CExt}(\beta) \wedge \text{CInt}(\beta) \wedge \text{After}_\beta \text{F}\varphi)) \\
& \leftrightarrow \text{Bel}_i ((\psi \rightarrow \text{Trust}(i, \alpha, \varphi)) \wedge (\neg \psi \rightarrow \text{Trust}(i, \beta, \varphi)))
\end{aligned}$$

#### 4.4 While

**Theorem 6.** *The formula*

$$\text{Trust}(i, (\text{while } \psi \text{ do } \alpha), \varphi) \leftrightarrow (\text{Bel}_i \text{After}_{(\psi?;\alpha)^*}(\psi \rightarrow (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha))) \wedge \text{Goal}(i, \varphi) \wedge \text{Bel}_i \text{After}_{\text{while } \psi \text{ do } \alpha} \text{F}\varphi \wedge \text{Bel}_i \text{F}\neg\psi)$$

is valid.

*Proof.* We have:

$$\begin{aligned} \text{Trust}(i, (\text{while } \psi \text{ do } \alpha), \varphi) &\leftrightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i \text{After}_{\text{while } \psi \text{ do } \alpha} \text{F}\varphi \wedge \\ &\quad \text{Bel}_i(\text{CExt}(\text{while } \psi \text{ do } \alpha) \wedge \text{CInt}(\text{while } \psi \text{ do } \alpha)) \\ &\leftrightarrow \text{Goal}(i, \varphi) \wedge \text{Bel}_i \text{After}_{\text{while } \psi \text{ do } \alpha} \text{F}\varphi \wedge \text{Bel}_i \text{F}\neg\psi \wedge \\ &\quad \text{Bel}_i \text{After}_{(\psi?;\alpha)^*}(\psi \rightarrow (\text{CExt}(\alpha) \wedge \text{CInt}(\alpha))) \end{aligned}$$

### 5 Application

Services-oriented architectures (SOA) allow to develop dynamic business processes and agile applications spanning across organisations and computing platforms to quickly adapt to ever changing requirements. By their modular nature, services can be composed to implement processes of various complexities.

Actors of SOA are divided into two roles, the client, having specific requirements, and the provider advertising its services. Non-functional parameters, such as quality of service (QoS) become important when selecting among a range of functionally equivalent services. However, in certain cases, discrepancies between advertised and observed QoS can occur, either because of temporary failures or voluntary over-rating from the provider. When facing such uncertainties, trust mechanisms should be used to select services matching the goals of the clients and providers.

Trust becomes even more crucial in composite services, where not only the client must trust the composite service but also where each provider involved in the composition must trust its partners [4]. Composite services can be modelled as a set of workflow patterns [10], which are equivalent to the complex action framework described in Section 4. Indeed, trust in a composite service depends on the services involved but also on the structure workflow. For example, a provider might agree to participate in a composite service if only its service is used at the end of a sequence, notably for data privacy concerns [3].

In the aforementioned paper, a multi-agent protocol is developed to entice providers to take part in composite web services. This protocol is centered around data privacy in composite services. Basically, and according to Theorem 4, a provider is willing to enter a composite service if and only if it trusts the providers of subsequent services to not mishandle its data. In other words the goal "not mishandle the data" only holds after its won service invocation thus fostering the need for trust.

### 6 Conclusion

We have presented in this work a logical formalization of trust in complex actions, and have sketched how this formalization could be useful for the formal characterization of

trust in composite services, where trust in a composed service is defined in a compositional way from trust in the components of that service. Directions of future research are manifold. In the present article we only gave a semantics for a logic of complex actions. On the one hand, future works will be devoted to find a complete axiomatization of the logic of Section 2 and to study the computational properties of this logic (decidability and complexity). On the other hand, we plan to extend the PDL-based formalism of Section 2 by parallel actions in order to be able to formalize services whose components might work in parallel.

## 7 Acknowledgements

We would like to thank the anonymous reviewers of LIS'2010 whose comments (hopefully) helped to improve the paper.

## References

1. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, Florida, October 1997.
2. Nuel Belnap, Michael Perloff, and Ming Xu. *Facing the Future: Agents and Choices in Our Indeterminist World*. Oxford University Press, Oxford, 2001.
3. Julien Bourdon and Toru Ishida. Trust chaining for provider autonomy in composite services. In *Joint Agent Workshop and Symposium (JAWS'09)*, 2009.
4. Julien Bourdon, Laurent Vercouter, and Toru Ishida. A multiagent model for provider-centered trust in composite web services. In *The 12th International Conference on Principles of Practice in Multi-Agent Systems (PRIMA 2009)*, number 5925 in LNAI, pages 216–228. Springer Verlag, 2009.
5. Cristiano Castelfranchi and Rino Falcone. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multiagent Systems (ICMAS'98)*, pages 72–79, 1998.
6. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
7. Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In C. Castelfranchi and Y. H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer, 2001.
8. J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time. *Journal of Computer and System Sciences*, 38:195–237, 1989.
9. J.Y. Halpern and J. H. Reif. The propositional dynamic logic of deterministic, well-structured programs. *Theoretical Computer Science*, 27:127–165, 1983.
10. Q. He, J. Yan, H. Jin, and Y. Yang. Adaptation of web service composition based on workflow patterns. *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC'08)*, Jan 2008.
11. Andreas Herzig and Dominique Longin. C&L intention revisited. In *Proc. 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 527–535. AAAI Press, 2004.
12. Andreas Herzig and Emiliano Lorini. A dynamic logic of agency I: STIT, abilities and powers. *Journal of Logic, Language and Information*, 19:89–121, 2010.

13. Andreas Herzig, Emiliano Lorini, Jomi F. Hübner, Jonathan Ben-Naim, Olivier Boissier, Cristiano Castelfranchi, Robert Demolombe, Dominique Longin, Laurent Perrussel, and Laurent Vercouter. Prolegomena for a logic of trust and reputation. In *3rd International Workshop on Normative Multiagent Systems (NorMAS 2008), Luxembourg, 15/07/2008-16/07/2008*, pages 143–157, [http://www.wen.uni.lu/fdef/luxembourg\\_business\\_academy/press](http://www.wen.uni.lu/fdef/luxembourg_business_academy/press), 2008. University of Luxembourg Press. ISBN: 2919940481.
14. Andreas Herzig, Emiliano Lorini, Jomi F. Hübner, and Laurent Vercouter. A logic of trust and reputation. *Logic Journal of the IGPL*, 18(1):214–244, February 2010. Special Issue “Normative Multiagent Systems”.
15. Emiliano Lorini. A dynamic logic of agency II: deterministic DLA, Coalition Logic, and game theory. *Journal of Logic, Language and Information*, 19(3):327–351, 2010.
16. Emiliano Lorini and Robert Demolombe. Trust and norms in the context of computer security. In *Proc. Ninth International Conference on Deontic Logic in Computer Science (DEON’08)*, number 5076 in LNCS, pages 50–64. Springer-Verlag, 2008.
17. Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.
18. E. Sirin, B. Parsia, and J. Hendler. Composition-driven filtering and selection of semantic web services. In *AAAI Spring Symposium on Semantic Web Services*, pages 129–138, 2004.
19. J. van Benthem and E. Pacuit. The tree of knowledge in action: Towards a common perspective. In G. Governatori, I. Hodkinson, and Y. Venema, editors, *Proc. of Advances in Modal Logic Volume 6 (AiML 2006)*, pages 87–106. College Publications, 2006.
20. Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, pages 150–157, Linköping, Sweden, 2003. IEEE Press.