Situation awareness and capacity in coalitions

Abstract—**In this paper, we propose a discussion on formal notions situation awareness and capacity in a coalition. .... We show the connection of a state-based logical model and an action-based logical model. The first model, named Interpreted systems is based on epsitimic logic notions and has been identified as a candidate for the formalisation of the situation analysis problem. The second model, named situation calculus, is a first-order logic of actions better suits to the formalisation of planning problems. We give here the definitions of both notions of (1) situation awareness and (2) capacity. The benefit of the connection between both models is twofold: First, it allows to introduce epistemic notions in capacities since most of the time an agent is capable of some action only if it knows its competences. Situation awareness can thus be seen as a condition to be met in order to validate a capacity of action. Seconldy, we can model the effect of the capacities of agents on their mental states, and the knowledge of capacities like "Agent i knows that it is able to ..., or "The group of agents G does not know that it can do ...". We thus define the notion of knowledge-based capacity.**

Keywords: Situation awareness, capacity, ability, coalitions, interpreted systems, situation calculus.

I. Introduction

- Higher-level information tasks like situation analysis require more logical-based appraoches [Kokar, Steinberg, Nous, Lambert,...]
- SA and planning task are difficultly dissociable
-
- Origin of coalitions
- Distinction with other teams: The adversary. The coalition is formed not only to bring together heterogeneous comptences, but also to beat another team or group.
- Even if we consider the static case of a coalition, a coalition is dynamic by nature.
- SAW in coalitions is materialized for example by a COP or a RAP...
- Roles!

II. Literature survey

A. Formalisms for coalition problems

- General problems: Formation, communication, coordination, cooperation
- A coalition is a set of cooperative agents in multi-agent systems.
- Research areas: Economy (game theory), sciences politiques (social choice theory), computer science (multi-agent systems and logical AI), military (information fusion...)
- Game theory: Notion of equilibria, formation of coalitions (members selection), communication (net)
- Central concepts: Commitment, cooperation, openning, competence, ability (Distinction between ability and capacity?) - Naturally modeled in an action-based formalism since closed to the planning task, necessary information to plan the mission, decide the coalition formation, the assets to be committed, etc.
- Related theories: Moore (knowledge and action), Cohen and Levesque (Intention), Rao and Georgeff (Bratman, BDI models), Singh, Werner, Wooldridge (multi-agent) [**?**], ??? (action logics like situation calculus), Fagin-Halpern-Moses-Vardi (epistemic logics)

Approach: Computer-science-based

B. Cooperation and coalition logics

Coalition logics allow to represent and reason about concepts such as ability, cooperation, commitment, goal, mission, plan, strategy, power, effectiveness ... In these logics , knowlegde representation is not well developed even if some attempts have been recently put forward [Wooldridge, ...]. Table of coalition logics and their features.

- Context
- Knowledge
- Strategy (proba.)

	Cooperation	Knowledge	Context	Observations
ATL [1]	$\times$			
ATEL [2]	$\times$	$\times$		
NATL [3]	$\times$		$\times$	
ATOL [4]	$\times$			$\times$
CL [5]	$\times$			
IS [**?**]		$\times$	$\times$	$\times$

TABLE I

Coalition logics.

In the last ten years, different logical formalisms have been proposed to deal with coalitions. Between 1997 and 2002, Alur, Henzinger, and Kupferman developed the Alternating-time Temporal Logic (ATL) [1], a logic of cooperative ability, allowing to express properties such as "Coalition $\mathcal{C}$ has the capability to bring about ϕ", formally denoted by $\langle\langle\mathcal{C}\rangle\rangle_\phi$. ATL is indeed based on the temporal logic Computation Tree Logic (CTL) in which the path quantifiers A (on all paths) and E (on some path) are replaced by cooperation modalities $\langle\langle\mathcal{C}\rangle\rangle_\phi$ meaning that "the group G can cooperate to ensure that ϕ" (or "the group G has a common strategy to force ϕ") [**?**].

ATL has been extended with knowledge modalities by van der Hoek and Wooldridge leading to Alternating-time Temporal Epistemic Logic (ATEL) [2]. In ATEL, some properties like "Coalition $\mathcal{C}$ can cooperate to bring about ϕ iff it is common knowledge in $\mathcal{C}$ that ψ".

In 2002, Pauly proposed a coalition logic [5] for reasoning about effectivity in game frames, and allowing expressions such as "Coalition $\mathcal{C}$ is effective for ϕ", where effective means that the coalition can guarantee that a given subset of states can be reach.

In 2005, ATL has been extended to account for constraints on the actions, thus introducing the notion of context [3]. The resulting logic is called Normative ATL (NATL), a logic of normative ability. In NATL, obligations and permissions can be introduced and the ability of the coalition takes then into account these constraints.

In [6], Goranko compares coalition logics of Pauly and alternating-time temporal logics of Alur *et al.* and comes to show that coalition game logics can be embedded into alternating-time temporal logics.

In 2006, Alternating-time Temporal Observational Logic (ATOL) has been proposed as an extension of ATL, designed to capture strategic properties of agents under incomplete information [4].

C. Information fusion standpoint

- SAW in coalitions implies notions of group knowlegde, of common understanding, constrained by an adversary, by a specific goal to reach, by cooperability constraints, uncertainty constraint (perception, reliability of sources...)
- SAW better model based state-based languages, like IS as proposed in [7]. This kind of model allows to explicitely represent situations, the environment, the uncertainty,
- Specific problem: how to conciliate both model (action-based and state-based)?
- Benefit sought: Define goals in terms of sets of states to reach, these states being possibly epistemic states like situation awareness is, thus validate formulas such as "Would agents of group G reach common awareness of ϕ before time t?"
- Alternative: (1) Introduce knowledge and uncertainty notions in SC [Bacchus etal, Scherl...] (2) Frame the action-based logic of situation calculus in a general state-based model like IS designed around knowledge representations, and extended to encompass quantified uncertainty concepts and game theory...
- Proposal: Follow way (2)! Already followed by Wooldridge that led to ATL and its friends!

The aim of this paper is thus to first present both languages of situation calculus and interpreted systems and draw the equivalences and disjunctions.

See modular interpreted systems in [8], also [9], or [2], [10], [11] for coalition epistemic logic.

III. Coalition as a distributed system

A. Interpreted systems

The interpreted systems language [12] is based on a relational semantics and provides a formal general framework for analyzing distributed systems through epistemic properties. In the following, we will use this language to formally represent and discuss the different types of vagueness through different mathematical models.

1) Background: Let us consider a set of agents $\mathcal{A} = \{1, 2, 3, \ldots, n, e\}$ where e is a special agent denoting the environment. Each agent is assumed to be in some *local state* l_i at a given time, encapsulating all the information the agent has access to. l_e denotes the local of the environment.

A *global state* s is an element of $S \subseteq L_1 \times \ldots \times L_n \times L_e$, where L_i is the set of the possible local states of the agent i. The local state of Agent i corresponds to the i^{th} component of the global state $s = (l_1, \ldots, l_n, l_e)$. A sequence of global states $s^1, s^2, \ldots$ is called a *run* r over S and is a function from time to global states. A *system* $\mathcal{R}$ is a set of runs. (r, m) denotes a point in $\mathcal{R}$, consisting of a run r and a time m. $r(m)$ represents the state of the system at time m. If $r(m) = (l_1, \ldots l_n, l_e)$ is the global state at point (r, m), then we define $r_e(m) = l_e$ and $r_i(m) = l_i$ for $i = 1, \ldots, n$ to be respectively the environment local state and the agents local states at point (r, m). A *round* m in run r is defined to take place between time $m - 1$ and time m.

Actions are the cause of changes in the system, and performed by the agents and the environment. Let ACT_i be the set of actions that can be performed by agent i, and let ACT_e be the set of actions performed by the environment. A *joint action* is an element of $ACT_e \times ACT_1 \times \ldots \times ACT_n$, *i. e.* a tuple $(\mathbf{a}_e, \mathbf{a}_1, \ldots, \mathbf{a}_n)$ of actions performed by the set of agents and the environment, where $\mathbf{a}_e$ is the action performed by the environment, and $\mathbf{a}_i$ is the action performed by the agent i.

A *protocol* P_i for agent i is a mapping from the set L_i of local states of the agent i to nonempty sets of actions in ACT_i. A protocol P_e for the environment e is a mapping from the set L_e of local states of the environment to nonempty sets of actions in ACT_e. Note that a protocol is a function on local states rather than on global states. A *joint protocol* P is a tuple $(P_1, P_2, \ldots, P_n)$ consisting of the protocols of each of the agents i, $i = 1, \ldots, n$. Note that P_e, the protocol of the environment, is not included in the joint protocol. Rather, the protocol of the environment is supposed to be given and P and P_e can be viewed as the strategies of opposing players.

A *context* γ is a tuple (P_e, S_0, τ, Ψ) where P_e is a protocol for the environment, S_0 is a nonempty subset of S describing the state of the system at the initiation of the protocol, τ is a transition function and Ψ is an admissibility condition on runs. τ is a *transition function*, which for each global state and each join action assigns the resulting global state of performing the join action. τ also describes which actions can be performed for a given global state. The *admissibility condition* Ψ on runs tells us which ones are "acceptable". Formally, Ψ is a set of runs and $r \in \Psi$ if and only if r satisfies the condition Ψ.

Note that the description of the behavior of a system is contextual, *i. e.* , a joint protocol P is always described within a given context γ.

Let Φ be a set of primitive propositions, describing basic facts about the system. Formulas are built using the classical operators of propositional logic. The set of formulas is closed off the operators $\neg$ and $\wedge$ (negation and conjunction). Hence, given two formulas ϕ and ψ, $\neg\phi$, $\phi\wedge\psi$, etc are also formulas. Let denote by $\mathcal{L}(\Phi)$ the language of Φ, *i. e.* the set of well-formed formulas.

An *interpreted system* $\mathcal{I}$ consists of a pair $\langle\mathcal{R},\pi\rangle$ where $\mathcal{R}$ is a system over a set S of global states and π is an interpretation for the propositions in Φ over S, which assigns truth values to the primitive propositions at the global states. Thus, for every $p\in\Phi$ and state $s\in S$, we have $\pi(s)(p)\in\{0;1\}$[1].

2) Knowledge and time: Blablabla

$$
\begin{aligned}
&(\mathcal{I},r,m)\vDash p \text{ iff } \pi(r,m)(p)=1\\
&(\mathcal{I},r,m)\vDash \phi\wedge\psi \text{ iff } (\mathcal{I},r,m)\vDash\phi \text{ and } (\mathcal{I},r,m)\vDash\psi\\
&(\mathcal{I},r,m)\vDash \neg\phi \text{ iff } (\mathcal{I},r,m)\nvDash\phi\\
&(\mathcal{I},r,m)\vDash K_i\phi \text{ iff } (\mathcal{I},r',m')\vDash\phi \text{ for all } (r',m')\\
&\qquad\qquad \text{such that } (r,m)\sim_i(r',m')\\
&(\mathcal{I},r,m)\vDash \bigcirc\phi \text{ iff } (\mathcal{I},r,m+1)\vDash\phi\\
&(\mathcal{I},r,m)\vDash \phi U\psi \text{ iff } \exists m'\geq m\\
&\qquad\qquad \text{such that } (\mathcal{I},r,m')\vDash\psi \text{ and } \forall m''\\
&\qquad\quad \text{such that } m\leq m''\leq m'.(\mathcal{I},r,m'')\vDash\phi
\end{aligned}
$$

where $\sim_i$ is an equivalence relation for agent i over the set of points of the system.

Add a discussion on complexity issues [13], [14].

See complexity of model checking temporal epistemic properties (van der Meyden).

3) Introducing uncertainty: Although this feature will not be detailed in this paper, we mention that the interpreted systems semantics can be extended to deal with uncertainty. In [15], the authors introduce probabilities on runs and thus define probabilistic systems. The probability measure can also be replaced by any other plausibility measure in the sense of [16] like belief measure, possibility measure, rank function, to name a few.

Add a discussion on complexity issues.

B. Mission

A protocol P_i can be seen as the strategy allowing agent i to achieve the mission $\mathcal{M}$. In general, it is possible to define a behavioral protocol which is a non-deterministic strategy to achieve the mission.

La mission M peut être vu comme une suite de buts ou états globaux du systèmes ordonnés de manière plus ou moins rigide. Ainsi la mission M peut être identifiée à un sous-ensemble de runs r, qui si matérialisé fera en sorte que la mission soit réalisé ou du moins potentiellement réalisable. On analyse ou conçoit donc le système distribué de façon à matérialiser, identifier, etc le sous-ensemble d'états globaux correspondants aux objectifs de la mission de la coalition.

Rajouter def de mission! La mission est une sorte de cahier des charges.

[1]0 states for False and 1 states for True.

C. Situation awareness in coalitions

1) Existing decision support concepts: - Definition - Example of COP, RAP, CTP... - Notions of coalition formation more of less dynamic linked to situation awareness. Pour l'instant, juste des display, mais rien pour raisonner.... d'ou necessite de truc! - Elements traites (track, positions,...) Introduire les notations...

2) Formalising situation awareness:

D. Ability in coalitions

Two ways of considering the ability notion:

1) Explicit modelisation (Formal definition either in SC or in IS), either related to action, either to protocol, or to state. (Make the link with know-how = competence, which can be encoded in the local state, or not)
2) Implicit modelisation (Describing the mission and compared to the joint protocol, and check the model)

IV. A Situation Calculus for Coalitions

A. Notions to be modelized

In our model, the mission is represented by a sequence of actions to be performed by the group of agents. In this case, a group succeeds in realizing a mission if it is able to perform the complex action which describes the mission.

The primitive notion on which ability is build is the notion of competence described as follows: competence represents the knowing-how of the agents relatively to an action. This knowledge may be inborn or may result from a learning phase. In our model, this information is considered as primitive. It must be noticed that this notion of competence is different from the one Cohen and Levesque consider in [17], where, if an agent competent for a proposition p believes p, then p is true.

From the notion of competence, we first define the notion of theoretical ability as follows:

Definition 4.1: Let A be a non empty set of agents (possibly a singleton), and a be a primitive action. A is theoretically able to perform a if:

1) A is competent to perform a
2) some conditions related to the agents of A are true

The conditions expressed in point 2 concern the agent (its physical state for instance, but not all the environment).

The notion of ability is finally defined as follows:

Definition 4.2: Let A be a non empty set of agents (possibly a singleton), and a be a primitive action. A is able to perform a if:

1) A is theoretically able to perform a
2) a is possible

B. A model of ability in the Situation Calculus

We suggest to use the Situation Calculus for two reasons:

- firstly, this formalism is a good candidate for modelling actions since it offers means to explicitly express preconditions and effects of actions;

- secondly, an important problem underlying this present work, the frame problem (i.e, how to express what are the changings induced by the performance of an action by an agent and how to express what remains unchanged), has been provided a solution in the Situation Calculus by Reiter.

1) The language: We consider a first order language $\mathcal{L}_{CS}$ which will allow us to model and reason about actions and ability. In this language, the changes of the world are resulting from action performances. It is defined as follows:

- a set of constants to represent agents.
- a set of functions and constants used to represent primitive actions, with parameters or without.
- a unary predicate $primitive(.)$ used to list the primitive actions.
- a binary function ; used to represent the sequence of actions.
- a constant S_0 used to represent the initial situation.
- a ternary function do. Here, unlike the "classical" Situation Calculus, the agent is not a parameter of the function which represents the action, but is a parameter of the function do which represents the performance of the action.
- a set of predicates called relational *fluents* which represent properties which may be changed by the performance of an action. The last argument of a fluent is a situation.
- a particular binary *fluent* $Poss$ used to express that an action is possible in a situation.
- a particular binary *fluent* is $competent$ and is used to represent the fact that an agent (or a group) is competent for performing a primitive action.
- a particular ternary *fluent* is $able\_t$ and is used to represent the fact that an agent (or a group) is theoretically able to perform an action.
- a particular ternary *fluent* is $able$ and is used to represent the fact that an agent (or a group) is able to perform an action.

2) The axioms: First, the initial state of the world must be represented. For doing so, for any fluent f and for any tuples $t_1, \ldots, t_n$ of ground terms such that $f(t_1, \ldots, t_n)$ is true in the initial situation, we consider the following axiom:

$$f(t_1, \ldots, t_n, S_0) \tag{1}$$

In particular, since $competent$ is a fluent, for any group G competent for performing the primitive action α in the initial situation S_0, we consider the following axiom:

$$competent(G, \alpha, S_0) \tag{2}$$

For any primitive action α, we consider an axiom of the following form:

$$primitive(\alpha) \tag{3}$$

We represent the preconditions of the primitive actions (i.e., the conditions that make the performance of the action possible) by an axiom of the following type:

$$\forall \alpha \forall S \; Poss(\alpha, S) \leftrightarrow pre(\alpha, S) \tag{4}$$

We then extend this kind of axioms for a sequence $\alpha;\beta$ where α is a primitive action and β is a complex action as follows:

$$\forall S \forall G \forall \alpha \forall \beta \; Poss(\alpha, S) \wedge Poss(\beta, do(G, \alpha, S)) \leftrightarrow Poss(\alpha;\beta, S) \tag{5}$$

Axiom (5) expresses that $\alpha;\beta$ is possible in S iff α is possible in S and β is possible after the performance of α in S.

Following Reiter [?], for any *fluent* $f(t_1, \ldots, t_n)$, we consider a successor state axiom which specifies all the ways the value of the fluent may change.

$$\forall S \forall G \forall \alpha \;\; Poss(\alpha, S) \rightarrow f(t_1, \ldots, t_n, do(G, \alpha, S)) \leftrightarrow \tag{6}$$
$$\gamma_f^+(t_1, \ldots, t_n, \alpha, S) \vee (f(t_1, \ldots, t_n, S) \wedge \neg \gamma_f^-(t_1, \ldots, t_n, \alpha, S))$$

$\gamma_f^+(t_1, \ldots, t_n, \alpha, S)$ represents the conditions which make f true after α has been performed in S. $\gamma_f^-(t_1, \ldots, t_n, \alpha, S)$ represents the conditions which make f false after α has been performed in S.

For any primitive action α, we consider an axiom of the following form:

$$\forall G \forall S \;\; competent(G, \alpha, S) \wedge conditions\_t(G, \alpha, S) \rightarrow able\_t(G, \alpha, S) \tag{7}$$

It expresses that a group G is theoretically able to perform α in situation S if G is competent for α in S and if some conditions related to G and α are satisfied.

Finally, in order to derive the theoretical ability for a group of agents, we consider:

$$\forall G \forall G' \forall \alpha \forall S \;\; primitive(\alpha) \wedge (G' \subseteq G) \wedge able\_t(G', \alpha, S) \rightarrow able\_t(G, \alpha, \tag{8}$$

$$\forall G \; G' \; \alpha \; \beta \; S \;\; (G' \subseteq G) \wedge able\_t(G', \alpha, S) \wedge able\_t(G, \beta, do(G', \alpha, S)) \rightarrow a \tag{9}$$

Axiom (9) expresses the fact that if a sub group G' of G is theoretically able to perform a primitive action α, then the group G is also theoretically able to perform α. Axiom (10) expresses that if a sub-group G' of G is theoretically able to perform α and if G is theoretically able to perform β once G' has performed α, then G is theoretically able to perform $\alpha;\beta$ (i.e., to perform α then β).

Finally, the following axiom allows to derive the ability of a group:

$$\forall G \forall \alpha \forall S \ able\_t(G, \alpha, S) \wedge Poss(\alpha, S) \rightarrow able(G, \alpha, S) \tag{10}$$

V. Connecting situation calculus and interpreted systems

In this part, we discuss the connection between the situation calculus and interpreted systems. As previously discussed in [18], situation calculus is not fundamental and the interpreted systems framework is even richer.

A. Preliminaries

Indeed, the common basis of both language is a transition states graph in which nodes are possible states and linked are labelled by actions. Since situation calculus is a logic of action, it reasons on sequences of actions (situations) and the graph is not explicitely expressed, the states being rather considered as transitory between actions. On the other hand, in the interpreted systems formalism which is mainly designed for reasoning on epistemic states, actions are only means for changing states. These two views can be seen as dual one of the other and in any case incompatible.

At a basis of the connection between situation calculus and interpreted systems is the mapping between the notions of "global state" in interpreted systems and of "situation" in situation calculus:

$$s \longleftrightarrow (l_e, l_1, \ldots, l_n)$$

Situation calculus is based on a first-order logic, whereas interpreted systems are described in a classical propositional language. However, this simplification is only for the ease of the exposition since as explained in [19], the interpreted systems language can be extended to include first-order predicates.

In the general case of a joint action $\mathbf{a} = [a_1, \ldots, a_n, a_e]^T$, the predicate do is defined by:

$$do(\mathbf{a}, s) = s' \tag{11}$$

where $s = (l_1, \ldots, l_n, l_e)$ and $s' = (l'_1, \ldots, l'_n, l'_e)$, l'_i being the resulting local state of achieving $\mathbf{a}$ from s, $i \in \mathcal{A}$. τ describes some prior knowledge about the world's behavior: To each global state, and to each action, it assigns the successor global state:

$$\tau(s)(\mathbf{a}) = s' \tag{12}$$

The pre-conditions of action $\mathbf{a}$ can then be deduced from τ. Indeed, if S_a is the set of all possible predecessor global states for action $\mathbf{a}$, the pre-conditions of $\mathbf{a}$ is represented by the formula of corresponding to this set of states.

A y re-reflechir!!!! Voir travaux de Wooldridge et Rao et Geogeff a propos des architectures BDI.

Table V-A summarizes the connections and differences between both languages.

Although some trivial similarities can be drawn between SC and IS, a major distinction between both languages requires more caution. Indeed, contrary to IS, the distributed aspect is not explicitly described in SC. In particular, when it is referred to a group of agents, the later is not detailed neither with the identifiers of the agents, neither with their local states, nor the actions for which they are competent. In the following two sections we discuss a possible extension of SC for distributed systems, *i. e.* a connection from SC to IS (Section V-B) and the coalition formation in IS, *i. e.* the more trivial connection from IS to SC (Section V-C).

Situation Calculus	Interpreted Systems	Notation
Set of agents	Set of agents	$\mathcal{A}$
-	Agent	i
Situation	Global state	s
-	Local state	l_i
Initial situation	Initial global state	s_0
Basic propositions	Basic propositions	Φ
do	Protocol	P_i
Pre- and post-conditions	Transition function	τ
Fluent	-	
-	Point	(r, m)
Sequence of actions	Run	r
-	Context	γ

TABLE II

Comparison between situation calculus and interpreted systems.

B. Situation calculus for distributed systems

The problem is addressed here from the standpoint of actions. We assume that an action is a joint action by default, and as such performed by more than a single agent. In SC, joint actions cannot be split into more elemenary actions whereas in IS joint actions are built from individual elementary actions and are thus a vector of actions $\mathbf{a} = [a_1, \ldots, a_n, a_e]^T$. In IS the notion of joint actions has been introduced to model concurrent actions [20]. Although concurrent actions have not originally considered in the situation calculus, this issue has been recently discussed in [21].

1) Different kinds of actions: In the field of action logics and thus in SC, different kinds of actions are considered **??**:

- Action composition $a_1; a_2$
- Repetition a^*
- Nondeterministic choice $a_1 | a_2$
- Concurrent actions $a_1 || a_2$
- Test action $p?a$

a) Test actions: Test actions are also called conditional actions [22]. In ISS (interpreted systems semantics), actions are performed according to a protocol P_i, which is defined for each agent as a function of its current local state to a subset of actions of ACT_i, $P_i : L_i \rightarrow 2^{ACT_i}$. A protocol is then a series a test actions of the form:

case of
  if t_1 **do** A
  if t_2 **do** B
end case

where t_1 et t_2 are standard tests of the form of a propositional formula of the language $\mathcal{L}(\Phi)$, are evaluated through the interpretation function π. By default, A is a nondeterministic action, *i. e.* a subset of ACT_i.

b) Action composition: Any change in the global state of the system results from an action execution, and thus between two actions exists a new global state for the system. Consequenlty, two different ways of modeling an action composition exist: (1) $s \stackrel{a_1;a_2}{\rightarrow} s'$ or (2) $s \stackrel{a_1}{\rightarrow} s'_1 \stackrel{a_2}{\rightarrow} s'$ and her are the two corresponding protocols:

case of
  if t_1 **do** $a_1; a_2$
end case

and

case of
  if t_1 **do** a_1
  if t_2 **do** a_2
end case

where t_2 corresponds to the post-condition of action a_1, and the agent will performed action a_2. A repetition action is simply a special case of a composition action.

c) Nondeterministic choice: If a protocol contains a nondeterministic choice, it is called a nondeterministic protocol.

d) Concurrent actions: Concurrent actions are intrinsically modelled in IS by the corresponding vector of joint action $\mathbf{a} = [a_1, \ldots, a_n, a_e]$.

2) Uractions: Let $A_0 = \{\Lambda, \mathsf{a}, \mathsf{b}, \ldots, a_1, a_2, \ldots\}$ be a set of basic joint actions for the n agents, where Λ is the null action. Two kinds of elements of A_0 can be distinguished:

1) Actions $\mathsf{a}, \mathsf{b}, \ldots$ that can be performed by a single agent (atomic joint actions);
2) Actions $a_1, a_2, \ldots$ that can only be performed by at least 2 agents (atomic joint actions).

Since in the ISS agents are the basic building blocks of the systems, we must find a way to artificially split an atomic joint action so that the vector a can be build. We thus introduce the notion of *uraction*. An uraction is a urelement of the set of actions[2]. As a urelement, a uraction is not an action by itself but must be an element of an action. This should not be confounded with the null action Λ. Figure 1 illustrates the space of joint actions for three atomic actions in $A_0 = \{\Lambda, \mathsf{a}, a_1, a_2\}$. Λ is the null action, a is an atomic joint action, *i. e.* which exists only if performed by two agents, and a_1 and a_2 are two atomic individual actions. In Figure 1, only 8 joint actions are possible for the 2 agents.

To A_0 is associated the set A'_0 containing the uractions corresponding to the atomic joint actions of A_0. In the example of Figure 1, $A'_0 = \{\Lambda, a', a_1, a_2\}$ where $\mathsf{a} = [a'; a']^T$.

3) Competence: Let A_0 be a set of basic joint actions for the n agents of a set $\mathcal{A}$. We define a competence function from A_0 to $\mathcal{P}(\mathcal{P}(\mathcal{A}))$ such that:

$$\begin{aligned} \mathrm{comp}: \quad & A_0 \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{A})) \\ & \mathsf{a} \mapsto \mathrm{comp}(\mathsf{a}) \end{aligned}$$

[2] "In set theory an urelement is something which is not a set, but may itself be an element of a set."

Fig. 1. Set of possible joint actions.

where $\mathrm{comp}(\mathsf{a})$ is the set of groups of agents having the competence for executing a. Thus, that means that the group of agents G is *competent* for executing the action a if and only if $G \in \mathrm{comp}(\mathsf{a})$.

Then the *competence* fluent can be derived:

$$\begin{aligned} competence: \quad & \mathcal{P}(\mathcal{A}) \times A_0 \rightarrow \{0;1\} \\ & (G, \mathsf{a}) \mapsto \begin{cases} 1 \text{ if } G \in \mathrm{comp}(\mathsf{a}) \\ 0 \text{ if } G \notin \mathrm{comp}(\mathsf{a}) \end{cases} \end{aligned}$$

In the example of Figure 1, if no other restriction is imposed, we have that $\mathrm{comp}(a_1) = \mathrm{comp}(a_2) = \{1, 2, (1,2)\}$ and $\mathrm{comp}(\mathsf{a}) = (1, 2)$.

4) Ability: Now we can go back to Definitions **??** and **??** of Section that can be re-stated as follows:

Definition 5.1 (Theoretical ability 2): Let $\mathcal{A} = \{1, 2, \ldots, k\}$ be a non-empty set of agents and let $\mathbf{a} = [a_1, a_2, \ldots, a_k]^T$ a joint action with $a_i \in A_0$. $\mathcal{A}$ is *theoretically able* to perform $\mathbf{a}$ if:

1) $\mathcal{A}$ is competent to perform $\mathbf{a}$, that is:

$$competence(\mathcal{A}, \mathbf{a}) = 1 \tag{13}$$

2) some conditions related to the agents of $\mathcal{A}$ are true.

In this case, the conditions lie the current situation concerning the agent (that it is not tired, for exemple). In IS, this property is encoded in the state of the environment, unless the agent has some information about its state of fatigue in which case this will be encoded in its local state. We return to this point in Section VI.

Definition 5.2 (Ability 2): Let $\mathcal{A}$ be a non-empty set of agents, let $\mathbf{a} = [a_1, a_2, \ldots, a_k]^T$ a joint action with $a_i \in A_0$ and let s be the current state of the system (*i. e.* the current situation). $\mathcal{A}$ is *able* to perform $\mathbf{a}$ if:

1) $\mathcal{A}$ is theoretically able to perform $\mathbf{a}$;
2) $\mathbf{a}$ is possible, that is:

$$\exists s \text{ such that } \tau(\mathbf{a})(s) \text{ is defined.} \tag{14}$$

This definition refers to

We can even introduce another definition of ability which takes into account the context. Indeed, something like, given a protocol P for agent i jointly executed in a context γ, the group of agents can be theoretically able to perform $\mathbf{a}$, able to perform $\mathbf{a}$ but never able to perform $\mathbf{a}$ since no run in the system leads to the state s, the pre-condition of $\mathbf{a}$.

C. Coalition formation in interpreted systems

Here we discuss the more trivial connection from IS to SC. Given a set of agents, each assigned with a set of actions for which it is competent, say ACT_i. In this direction, there is no incompatibility between SC and IS. Indeed, ...

VI. Awareness and ability

A. Awareness as a precondition of action

Adding a knowledge condition into the definitioin of capacity.

Directly linked to knowledge-based protocols.

Example of coordinated action.

B. Capacity of knowing

Link to epistemic actions. 1. Capable of executing a.

2. $[s]\mathsf{a} = s'$ and $(\mathcal{I}, r, m) \vDash K_i\phi$

See security protocols. link between permission and capacity.

C. Knowing capacity

Knowing that a group of agent is able to execute a.

$\phi = capable(G, \mathsf{a})$ and verify if $(\mathcal{I}, r, m) \vDash K_G\phi$

Awareness is the computationable knowledge. Agent i is aware of ϕ ($A_i\phi$) if it is capable of verifying $K_i\phi$.

VII. Conclusions

On a tous les morceaux pour faire de la formation de coalitions : (1) fabrication des groupes d'agents, (2) raisonner sur les capacites et les competences. Interessant: Lies ces notions et SAW. On est dans la situation ou on invite un agent a rejoindre la coalition. L'agent et capable d'analyser ses capacites et comptences mais refuser de joindre la coalition et donc de se commettre. Donc, si un agent est capable et comptent il peut refuser de se commettre... a joindre la coalition. (Attention, discussion de niveau superieur) Avec les memes notions, on peut adapter la discussion pour entrer la dynamique dans la patente!

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman, "Alternating-time temporal logic," *Journal of ACM*, vol. 49, no. 5, pp. 672–713, 2002.

[2] W. van der Hoek and M. Wooldridge, "Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications," *Studia Logica*, vol. 75, pp. 125–157, 2003.

[3] M. Wooldridge and W. van der Hoek, "On obligations and normative ability: Towards a logical analysis of the social contract," *Journal of Applied Logic 3 (2005)*, vol. 3, pp. 396–420, 2005.

[4] W. Jamroga and T. Ågotnes, "What agents can achieve under incomplete information," in *AAMAS'06*, Hakodate, Hokkaido, Japan, May 8-12 2006.

[5] M. Pauly, "A modal logic for coalitional power in games," *Journal of Logic and Computation*, vol. 12, no. 1, pp. 149–166, 2002.

[6] V. Goranko, "Coalition games and alternating temporal logics," in *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge (TARK)*, 2001, pp. 259–272.

[7] P. Maupin and A.-L. Jousselme, "A general algebraic framework for situation analysis," in *Proceedings of the 8^{th} International Conference on Information Fusion*, Philadelphia, PA, USA, July 2005.

[8] W. Jamroga and T. Ågotnes, "Modular interpreted systems: A preliminary report," Department of Informatics, Clausthal University of Technology, Tech. Rep. IfI-06-15, 2006.

[9] V. Goranko and W. J. Jamroga, "Comparing semantics of logics for multi-agent systems," *Synthese*, vol. 139, no. 2, pp. 241–280, 2004.

[10] W. van der Hoek and M. Wooldridge, "Model checking knowledge and time," in *Proceedings of the 9th International SPIN Workshop - Model Checking of Software*, Grenoble, France, April 11-13 2002.

[11] W. van der Hoek, M. Wooldridge, and S. van Otterloo, "Model checking knowledge and time via local propositions: Cooperative and adversarial systems," 2004.

[12] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about knowledge*. Cambridge, MA: The MIT Press, 2003.

[13] J. Y. Halpern, R. van der Meyden, and M. Vardi, "Complete axiomatizations for reasoning about knowledge and time," *SIAM J. Computing*, vol. 33, no. 3, pp. 604–703, 2004.

[14] J. Y. Halpern and Y. Moses, "A guide to completeness and complexity for modal logics of knowledge and belief," *Artificial Intelligence*, vol. 54, pp. 311–379, 1992.

[15] J. Y. Halpern and M. Tuttle, "Knowledge, probability and adversaries," *Journal of the Association for Computing Machinery*, vol. 40, no. 4, pp. 917–962, 1993.

[16] N. Friedman and J. Y. Halpern, "Plausibility measures: A user's guide," in *Proceedings of the Eleventh Conference on Uncertainty in AI*, 1995, pp. 175–184.

[17] P. R. Cohen and H. J. Levesque, "Intention is choice with commitment," *Artificial intelligence*, vol. 42, no. 2-3, pp. 213–261, 1990.

[18] F. Bacchus, J. Halpern, and H. Levesque, "Reasoning about noisy sensors and effectors in the situation calculus," *Artificial Intelligence*, vol. 111, no. 1-2, pp. 171 – 208, 1999.

[19] J. Y. Halpern, *Reasoning about Uncertainty*. Cambridge, MA: The MIT Press, 2003.

[20] J. Y. Halpern and R. Fagin, "Modeling knowledge and actions in distributed systems," *Distributed computing*, vol. 3, no. 3, pp. 159–177, 1989.

[21] R. B. Scherl, "Reasoning about the interaction of knowledge, time and concurrent actions in the situation calculus," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, August 9-15 2003, pp. 1091–1096.

[22] B. van Linder, W. van der Hoek, and J.-J. C. Meyer, "Tests as epistemic updates - Pursuit of knowledge," Institute of Information and Computing Sciences, Utrecht University, Tech. Rep. UU-CS-1994-08, 1994.