Distributed Cognition through Active Mental Entities: an Argumentation-Based Approach

P. Baroni, D. Fogli, M. Giacomin, G. Guida
Università di Brescia, Italy

This paper introduces the active mental entities approach to the design of intelligent autonomous agents. In this approach, mental attitudes are conceived as active computational entities and the overall mental activity of an agent results from their independent operation and interaction. In order to provide a formal support to this approach, we propose argumentation theory as a basis for the definition of interaction mechanisms among active mental entities. To this purpose, a novel distributed argumentation algorithm is devised. Finally, we give some operation examples in the context of a robotic application.

Keywords: agent architecture, distributed cognition, mental attitudes, argumentation, conflict resolution.

1. Introduction

A major problem in the field of autonomous agents is how an agent with limited resources can cope with two often conflicting requirements:

– sensing, reasoning, and acting in accordance with the real-time constraints imposed by the environment;

– maintaining a behavior coherent with agent motivations and oriented to the achievement of endogenous goals (Castelfranchi, 1995).

For the satisfaction of the above requirements, this paper proposes an innovative approach to model and design autonomous agents, which is based on two main standpoints.

1 We thank the anonymous referees and the guest editors for their insightful comments and suggestions.

2 Address for correspondence: Prof. Pietro Baroni, Dipartimento di Elettronica per l’Automazione, Università di Brescia, Via Branze 38, 25123 Brescia, Italy. E-mail: baroni@ing.unibs.it
Active mental entities as basic components of agent cognitive activity. In several approaches to agent systems, mental attitudes such as beliefs, intentions, and goals, are considered as passive data structures, managed by an algorithm that must account for all aspects of agent’s behavior. We suggest that a more effective approach to model the cognitive activity of an agent can be founded on the concept of active mental entities, understood as concurrent processes capable of autonomously evolving and interacting. At any time during agent operation, there is a set of active mental entities which are running and that determine the cognitive activity of the agent. Active mental entities combine a distributed organization, which is suitable to ensure prompt reactivity to external stimuli, with explicit motivations, which are often ruled out in purely reactive systems.

Distributed argumentation as the basic computational model of interaction among active mental entities. In order to ensure the coherency of the behavior emerging from the interactions among active mental entities, a suitable theoretical framework is needed. In our proposal we rely for this purpose on argumentation theory. As extensively discussed in the literature, e.g. (Dung, 1995; Pollock, 1990), argumentation is appropriate to model practical defeasible reasoning; it can cope with partial and uncertain information and can take into account resource-boundedness and time constraints. Moreover argumentation is based on a notion of conflict resolution, which has also a central role in governing interaction among mental entities. However, since active mental entities are conceived as concurrent processes, argumentation theory has to be extended to cope with distributed processing.

The paper is organized as follows. In Section 2 a novel distributed argumentation algorithm is introduced. Section 3 presents the notion of active mental entity and describes our approach to the specification and design of the cognitive activity of an agent. Section 4 provides some examples that demonstrate the applicability and the advantages of the proposed approach with reference to the operation of a waiter robotic agent, inspired to the AAAI robot competition. Section 5 provides a comparison with related works, and Section 6 concludes the paper.

2. Distributed argumentation

Argumentation is a framework for reasoning with incomplete and uncertain information, which models non-monotonic reasoning as the process of constructing and comparing arguments for propositions. Arguments represent composite reasons for believing propositions, and are constructed from a given set of premises, by chaining rules of inference to reach a conclusion. A distinctive feature of argumentation is that premises can be uncertain and the rules of inference may represent just provisional reasons for their conclusions, that can be defeated in presence of new information. In such a
framework, different arguments may contradict each other; therefore it is possible that previously accepted propositions are discarded in front of new emerging counter-arguments. To this aim, an order relation among arguments has to be defined, reflecting their relative strength or conclusive force (Pollock, 1991). Conflicts among arguments are expressed by a binary relation, called defeat relation: given two arguments α and β, α is a defeater of β if α conflicts with β and α is not weaker than β. Given a set of arguments, the core problem consists then in computing their defeat status, that is determining which ones of them turn out to be undefeated from the conflict: the conclusions of the undefeated arguments are the most credible ones and should be believed.

As pointed out by Pollock (1992), argumentation provides a natural way to cope with the limitation of physical and computational resources inherent to any real agent. On one hand, the construction of arguments can be interest-driven (Pollock, 1990), namely focused on finding arguments and counter-arguments only for propositions of interest for the agent. On the other hand, the reasoning activity itself is not forced to be complete: at any time the agent has a set of justified conclusions, but some of them might be discarded at a later stage of the reasoning and new ones may emerge (Pollock, 1992). As a consequence, the agent has, at any moment, the option either of acting on the basis of the available (provisional) conclusions or of continuing the reasoning process to derive new and possibly more informed conclusions. The construction and use of arguments as a paradigm for practical reasoning can be considered as a method for handling the logical uncertainty emerging from facts that do not necessarily need to be composed all together into a coherent and consistent frame (Elvang-Gøransson et al., 1993). Argumentation has been proposed as a logical foundation of intelligent reasoning both at the epistemological level (i.e. reasoning on what to believe) and at the practical level (i.e. reasoning on what to do). As to the former level, we have shown in (Baroni et al., 2000) that it provides an integrated model for belief revision and defeasible reasoning, as to the latter, it has been proposed in (Pollock, 1998) as an effective logical foundation of defeasible planning.

In order to carry out argumentation, the agent is engaged in two main activities: on one hand, it has to develop inferences on the basis of premises continuously updated by perceptual activity and of the conclusions previously drawn; on the other hand, it has to continuously revise the defeat status of its conclusions as a consequence of the new arguments produced by the inferential activity. In all the argumentation approaches we are aware of, these two activities are delegated to a centralized algorithm. These approaches are inherently unable to exploit the potential advantages of a distributed organization, advocated by several authors, e.g. (Brooks, 1986; Firby et al., 1995), as the most appropriate one for the architectural design of
autonomous agents: the evolution of autonomous agent architectures has pushed forward distribution as a proper organizational principle, both at the level of logical computation and of processing resources. As it will be discussed in next section, our proposal extends this organizational principle to the level of cognitive activity. In the field of autonomous agents it seems therefore interesting to investigate the combination of the above mentioned promising research lines: argumentation, as a well-founded model of practical reasoning, and distributed organization, as an architectural principle. To this purpose, a distributed argumentation algorithm is needed, where independent processing nodes communicate and cooperate to produce arguments and compute their defeat status.

Moreover, since agents operate in a dynamic world, the set of arguments and their defeat relationships are dynamic as well, i.e. new arguments can be generated and old ones can be removed as a consequence of inferential and perceptual activity. In this context, an important requirement is that, after a modification in the set of arguments has occurred (and therefore their defeat relationship has changed), the distributed algorithm is able to reach, in a finite amount of time, a termination state, where the arguments are assigned the same defeat status they would be assigned by a centralized algorithm exploiting global information. This requirement is equivalent to the property of self-stabilization, which states that starting from an arbitrary and possibly illegal initial state, the system always returns to a legal state in a finite number of steps (Dolev, 2000). Self-stabilization ensures that distributed argumentation produces coherent results even in absence of any high-level, centralized coordination. A distributed self-stabilizing algorithm for argumentation has been introduced in (Baroni & Giacomin, 2001), based on the general assumption that for each argument there is exactly one process managing it. The results obtained under this general assumption are guaranteed to hold also in realistic contexts, where a process may be allowed to manage several arguments. Here we present an extended version of this algorithm, adapted to the needs of the architecture that we will describe in next section.

In order to illustrate our algorithm, we have to introduce the definition of legal state, that corresponds to specifying the semantics of the argumentation framework. Among the various kinds of semantics proposed in the literature, the most suitable for a distributed environment seems to be the grounded semantics (Dung, 1995): in the following, we recall the relevant definition as it has been introduced in (Pollock, 1992), i.e. by referring to the notion of level. Given a set of arguments $A$ partially ordered by a binary defeat relation, it is possible to identify for each level a relevant set of arguments:

- all arguments of $A$ are in at level 0;
an argument of \( A \) is in at level \( n+1 \) if and only if it is not defeated by any argument in at level \( n \).

According to this definition, the status of an argument is formally defined as follows:

- an argument is undefeated if and only if there is a level \( m \) such that for every \( n \geq m \), the argument is in at level \( n \);
- an argument is defeated if and only if there is a level \( m \) such that for every \( n \geq m \), the argument is out at level \( n \);
- an argument is provisionally defeated if and only if there is no level \( m \) such that the argument is in at all higher levels or out at all higher levels.

Intuitively, the underlying idea is that:

- an undefeated argument is one which the agent should believe, since it has no defeaters or all its defeaters are in turn defeated;
- a defeated argument is one which the agent should disbelieve, since it has some undefeated defeaters and therefore lacks a valid support;
- a provisionally defeated argument is controversial, since its defeaters are in turn provisionally defeated: as an example, there may be two contradicting but equally believable arguments, so that one can not express a preference for either of them.

The goal of our distributed argumentation algorithm is to compute the defeat status of a set of arguments in a distributed context, without relying on any kind of global representation: the defeat status has to emerge from the autonomous activity and the interaction of individual nodes, which exploit local information about the status of their defeaters only. To this purpose, the self-stabilizing algorithm has to enforce a legal state in which the arguments are labeled as prescribed by the definition above.

First of all, let us introduce some more details about our model of argumentation. We assume that an argument, say \( \sigma \), for a proposition, say \( q \), can be produced in two ways:

- it can be directly acquired from an information source, e.g. a sensorial activity or an existing knowledge base; in this case it is called a premise;
- it can be inferentially derived from other existing sub-arguments \( \sigma_{i}, \ldots, \sigma_{n} \) for propositions \( p_{i}, \ldots, p_{n} \) by means of a rule of inference \( R(p_{i}, \ldots, p_{n} \Rightarrow q) \); in this case it is called a derived argument.

Since distinct arguments feature different credibility, it is useful to associate with each argument \( \sigma \) a measure of its uncertainty, called degree of strength and indicated by \( \text{strength}(\sigma) \). In the case of a premise, the degree of strength expresses the inherent uncertainty of the information source it is...
acquired from. In the case of a derived argument, both the uncertainty of the sub-arguments and of the rule of inference have to be taken into account. As in (Pollock, 1991), we suppose that a strength value $\text{strength}(R)$ characterizes the degree of certainty of a rule $R$, and we compute the strength of the argument according to the *weakest-link principle*, i.e. by taking the minimum among the strength of the rule and those of the arguments supporting its premises: $\text{strength}(\sigma) = \min\{\text{strength}(\sigma_1), \ldots, \text{strength}(\sigma_i), \text{strength}(R)\}$.

We can now define more precisely the notion of defeater. An argument $\sigma$ is a *defeater* of another argument $\tau$ if and only if:

- the conclusion of $\sigma$ contradicts the conclusion of a sub-argument $\tau_i$ of $\tau$ (including the case that $\tau = \tau_i$); and
- $\text{strength}(\sigma) \geq \text{strength}(\tau_i)$.

According to this definition, corresponding to the notion of rebutting defeat (Pollock, 1992), two types of defeaters for an argument $\tau$ are identified: direct defeaters, which deny exactly the conclusion of argument $\tau$; indirect defeaters, which undermine the construction of argument $\tau$, by denying one of the sub-arguments used in its derivation.

In our model, each process $\alpha$ is in charge of an argument, which in the sequel, for simplicity, will be identified with $\alpha$ itself. In order to determine its defeat status, $\alpha$ exchanges information with the processes associated with its direct defeaters (the set of these arguments is indicated by $\text{d-defeaters}(\alpha)$) and with the sub-arguments from which it is derived (the set of these sub-arguments is denoted by $\text{IMM}(\alpha)$, of course $\text{IMM}(\alpha) = \emptyset$ if $\alpha$ is a premise).

The set $\text{d-defeaters}(\alpha)$ can be partitioned in turn into two classes: *superiors*(\alpha), which have a strength strictly greater than $\alpha$; *contenders*(\alpha), which, having the same strength as $\alpha$, are defeated by $\alpha$ in turn.

The basic idea of our algorithm is the following. Each process $\alpha$ continuously monitors the defeat status of its direct defeaters and of its sub-arguments, and every time $\alpha$ detects a change, it updates its own defeat status $s[\alpha]$ according to the following rules:

- if $\exists \beta \in \text{IMM}(\alpha) : s[\beta] = \text{DEF}$ then $s[\alpha] = \text{DEF}$
- if $\exists \beta \in \text{IMM}(\alpha) : s[\beta] = \text{PROV}$ and $\forall \gamma \in \text{IMM}(\alpha) s[\gamma] \neq \text{DEF}$ then
  - if $\exists \beta \in \text{superiors}(\alpha) : s[\beta] = \text{UNDEF}$ then $s[\alpha] = \text{DEF}$
  - otherwise $s[\alpha] = \text{PROV}$
- if $\forall \beta \in \text{IMM}(\alpha) s[\beta] = \text{UNDEF}$ or $\text{IMM}(\beta) = \emptyset$ then
  - if $\exists \beta \in \text{superiors} : s[\beta] = \text{UNDEF}$ then $s[\alpha] = \text{DEF}$
  - if $\exists \beta \in \text{superiors}(\alpha) : s[\beta] = \text{PROV}$ and $\forall \gamma \in \text{superiors}(\alpha) s[\gamma] \neq \text{UNDEF}$ then $s[\alpha] = \text{PROV}$
  - if $\forall \gamma \in \text{superiors}(\alpha) s[\gamma] = \text{DEF}$ or $\text{superiors}(\alpha) = \emptyset$ then
- If $\forall \gamma \in \text{contenders}(\alpha) s[\gamma] = \text{DEF}$ then $s[\alpha] = \text{UNDEF}$
- otherwise $s[\alpha] = \text{PROV}$

The correctness of the algorithm has been formally proved in (Giacomin, 2002): starting from an arbitrary initial state, the algorithm reaches in a finite amount of time the legal state, i.e. the one corresponding to the defeat status assignment prescribed by the grounded semantics. Moreover, this state is stable, in that every argument satisfies the above conditions and therefore each process $\alpha$ does not update its defeat status $s[\alpha]$ any more. We remark that this result holds without any synchronization assumption and without any centralized control: the global defeat status emerges from the activity of the individual processes by means of a suitable local information propagation mechanism.

3. Active mental entities

Cognitive activity is often described by using terms such as desires, beliefs, intentions, obligations, emotions, etc. These terms denote mental attitudes, i.e. entities that are inside the mind of an intelligent agent and which are responsible of its external behavior. Our main point concerns the fact that it is advantageous to model mental activity as a form of distributed computing. In a distributed computing framework, a set of autonomous processes execute distinct threads of control and interact in order to achieve some globally useful objectives. Clearly, a multi-agent system is a kind of distributed computing framework where each agent is regarded as an autonomous processing entity. We suggest that also the mental activity of a single agent can in turn be modeled in this way. In fact, if we consider the mental processes occurring in our mind, we can reasonably describe them as the result of the cooperation and conflict between various intentions, inhibitions, hopes, desires, etc. A similar description of human reasoning has been informally proposed by Minsky (Minsky, 1986).

Therefore, we model mental attitudes as active mental entities, namely as entities able to autonomously perform elementary cognitive tasks and to interact with each other. At any time in the agent life, a number of active mental entities operate concurrently and give rise to the current mental endowment of the agent. Active mental entities can be regarded as an evolution of the traditional concept of static mental attitudes. In a schematic view, we can consider three approaches to modeling mental attitudes. At a first level, mental attitudes can be regarded as static data structures, manipulated by a suitable algorithm that implements the overall mental activity of an agent. In this context, each type of mental attitude (e.g. belief, intention,…) roughly corresponds to a data type in a procedural programming language. If one adopts an object-oriented modeling approach, each type of mental attitude corresponds to a class, whose definition specifies both data struc-
tures and the methods operating on them. This approach may have significant advantages from a software engineering point of view, but adopts the same underlying centralized computational model. The active mental entities approach makes a step further by regarding mental attitudes as active objects (Booch et al., 1998): for each class of mental attitudes an independent thread of control is specified. Actual mental attitudes are instances of these classes, i.e. processes in a distributed computing framework, each one executing the thread of control specified in the class it belongs to. These processes are instantiated at run-time, depending on current needs, and are disposed when no more needed.

3.1 Designing an agent architecture based on active mental entities

An agent architecture can be defined as “a particular methodology for building agents. It specifies how the agent can be decomposed in the construction of a set of component modules and how these modules should be made to interact” (Maes, 1991). According to this definition, the active mental entities approach is the background on which a complete agent architecture can be developed. Our agent architecture is constituted by three layers:

a) a cognitive layer, including active mental entities;

b) an operative layer, including components in charge of performing actions, either physical, concerning the interaction with external world through sensors and actuators, or symbolic, such as computational activities;

c) a knowledge base, representing the basic agent competence endowment that can be exploited by all agent components (active mental entities and operative components) during their operation.

In the following we will mainly deal with layer a) which is the most significant in this context. Some more details on layers b) and c) are given in (Baroni & Fogli, 2000). In particular the design of the cognitive layer is based on the following methodology that will be discussed in next sections:

a.1) identify the classes of active mental entities necessary to model the cognitive tasks of the agent in the considered application domain;

a.2) for each class of active mental entities, specify its behavior, namely the thread of control to be executed by each instance of the class;

a.3) for each class, define the mechanisms of generation and disposal of an instance of the class;

a.4) for each class, specify the interactions that may occur between an instance of the class and other instances of the same or other classes.

3.2 Intenders and attenders

As to the step a.1), two classes of active mental entities, namely intenders and attenders, have been identified and used in our first architecture proposal. Intenders are related with the intention to do something, while attenders are related with the need to focus attention about some aspects of the world.
These two classes have been selected as a starting point since they are related to the most basic activities of an agent, namely sensing and acting -see also (Baroni & Fogli, 2000). In particular, intenders are devoted to proactively pursue a given goal by generating and executing proper action strategies and revising them, whenever appropriate. Attenders are in charge of forming beliefs about the world and keeping them up to date: they enable an agent to focus attention on interesting aspects of the world, either monitoring environment evolution or actively looking for useful evidences.

Intenders and attenders can be related either to the agent permanent needs, specified at design time, or to transient needs, relevant to the actual satisfaction of the permanent needs in a specific context. For instance, in our example concerning a waiter robotic agent, the permanent need of preserving the battery charged may give rise to the transient need of going to a recharging point, whenever the robot realizes that the energy level has reached a minimum threshold. The following subsections describe how steps a.2), a.3) and a.4) are realized for attenders and intenders.

3.3 Attenders

An attender is an active mental entity committed to form a belief about some interesting aspect of the world, i.e. to ascribe (and revise, whenever appropriate) a truth value to a proposition of interest for the agent. An attender is generated when a new interesting proposition is identified and is dismissed when the interest in the proposition ceases. In particular, primary attenders correspond to permanent information interests, such as knowing whether “battery charge level is over a given threshold”, and last for the whole agent life. Secondary attenders correspond to transient information needs and last only for the duration of the relevant agent interest.

An attender constructs arguments and counterarguments for the proposition S it is in charge of, and computes their defeat status in order to find the most believed truth value for S. More formally, an attender is defined as a six-tuple $A=(S, V, str, \Sigma_T, \Sigma_F, AM)$, where:

- $S$ is the subject of the attender, namely a proposition whose truth or falsity is of interest for the agent;
- $V$ is the currently most believed truth value of the subject $S$, where $V \in \{T,F,UNK\}$;
- $str$ is the degree of credibility of $V$, meant as the strength of the strongest argument supporting $V$; $str \in [0, 1] \cup \{\oplus\}$: $str$ is defined in the interval $[0, 1]$ of real numbers if $V \in \{T,F\}$, and takes the undefined value $\oplus$ otherwise;
- $\Sigma_T$ is the current set of arguments that support the truth value $T$ for $S$;
- $\Sigma_F$ is the current set of arguments that support the truth value $F$ for $S$;
AM is a set of methods used by the attender in its operation.

Initially, when an attender is generated, only S and AM are given; V and \(\text{str}\) are computed and revised dynamically by constructing arguments in \(\Sigma_T\) and \(\Sigma_F\). Each argument can be constructed in one of the following ways:

a) directly finding S (or \(\neg S\)) in the agent knowledge base;

b) collecting data from the external world, by means of sensorial activity, that provide immediate evidence for S (or \(\neg S\));

c) constructing an argument by means of a rule of inference available in the agent knowledge-base from which either S or \(\neg S\) can be derived.

In cases a) and b), the argument corresponds to a premise, coinciding with the subject of the attender or with its negation; the relevant strength value reflects the degree of credibility of the agent knowledge base in case a) or the reliability of the involved sensor and sensorial activity in case b). Since the external world is generally dynamic, the validity of an argument based on sensorial activity may vary over time or its strength may change; the attender has to check and, if appropriate, revise V and \(\text{str}\) and notify its generating mental entity accordingly.

In case c), the agent exploits its relational knowledge in order to find a rule with conclusion S or \(\neg S\). Since the premises of this rule become propositions of interest for the agent, the relevant attenders are generated. In particular, if the rule of inference is \(R=(p_1, \ldots, p_n \Rightarrow q)\), with \(q \in \{S, \neg S\}\), the attenders \(A_1=(p_1, V_1, \text{str}_1, \Sigma_{T1}, \Sigma_{F1}, \text{AM})\), \(\ldots\), \(A_n=(p_n, V_n, \text{str}_n, \Sigma_{Tn}, \Sigma_{Fn}, \text{AM})\) are generated, and they are committed to notify the generating attender every time a new argument is constructed or the defeat status of a previously constructed argument is revised. Using this information, the attender constructs the arguments in \(\Sigma_T\) and \(\Sigma_F\), and computes the relevant strength according to the weakest link principle. In such a framework, it is possible that one or more arguments support a conclusion S, while others support its negation \(\neg S\); the defeat status is computed on the basis of the distributed self-stabilizing algorithm presented in the previous section.

Given the sets of arguments \(\Sigma_T\) and \(\Sigma_F\) along with their computed defeat status, the most believed truth value V for S and its strength \(\text{str}\) have to be determined. The value UNK is assigned to V in three situations, namely:

- when a provisionally defeated argument is present in \(\Sigma_T\) or \(\Sigma_F\); in this case \(\Sigma_T \cup \Sigma_F\) does not include undefeated arguments in the stable state: intuitively, this occurs for instance when there are equally plausible conflicting arguments for both S and \(\neg S\);

- when all the arguments of \(\Sigma_T \cup \Sigma_F\) are defeated or \(\Sigma_T \cup \Sigma_F = \emptyset\); this case denotes the absence of any valid support for either S or \(\neg S\);

- when the defeat status is incompatible with the grounded semantics (e.g., when there are undefeated arguments both in \(\Sigma_T\) and \(\Sigma_F\)); this assign-
ment is therefore only temporary and will evolve in the right one as soon as the legal state is reached.

The truth value $T$ ($F$) is assigned to $V$ when there is an undefeated argument in $\sum_T$ ($\sum_F$) while all the arguments in $\sum_F$ ($\sum_T$) are defeated (it can be easily shown that, in the stable state, the presence of an undefeated argument in $\sum_T$ ($\sum_F$) entails that all the arguments of $\sum_T$ ($\sum_F$) are defeated). In this case, the degree of credibility of the assignment reflects the best reason supporting it; $str$ is assigned the strength value of the strongest undefeated argument in $\sum_T$ ($\sum_F$).

3.4. Intenders

An intender is an active mental entity committed to pursue a goal; it is generated when a new goal is identified and is dismissed when the goal is reached or aborted. In particular, primary intenders correspond to permanent goals, such as “preserving physical integrity” or “avoiding collisions with persons” and last for the whole agent life. Secondary intenders correspond to transient goals, such as “going to a recharging point”, are created by other intenders or attenders, and are disposed when they have no more reason to exist. More formally, an intender is defined as a four-tuple $I=\langle S, IC, P, IM \rangle$, where:

- $S$ is the subject for the intender, namely a proposition describing a goal or a desired state of the world;
- $IC$ is a triggering condition, which enables an intender to act in order to pursue its goal;
- $P$ is the current action plan that is supposed to achieve the subject $S$;
- $IM$ is a set of methods used by the intender in its operation.

Initially, when the intender is generated, only $S$, $IC$, and $IM$ are given; $P$ has to be constructed, executed, and possibly dynamically revised by the intender itself during operation. Therefore, the set of methods $IM$ must include all the procedural elements necessary to carry out these tasks. In particular, a generation method constructs candidate action plans (either through planning from first principles or resorting to a library of pre-compiled plans) suitable to achieve the subject $S$. A plan is a sequence of actions which can be elementary, consisting of computations, sensory data acquisitions or concrete actions on the environment, or non-elementary, corresponding to goals to be achieved, and therefore requiring the creation of new intenders.

Each plan $PL_i$ has an associated applicability condition $AC_i$ representing a condition under which the plan can be executed: $AC_i$ becomes therefore an interesting proposition for the selection method, in charge of selecting $P$ among a set of candidate plans. As a consequence, in correspondence to each candidate plan $PL_i$ with applicability condition $AC_i$, an attender $A_i=\langle AC_i, V_i, str_i, \sum_{ti}, \sum_{fi}, AM \rangle$ is generated. The selection method, exploiting the informa-
tion provided by the generated attenders (the truth value of $AC_i$ and the relevant degree of credibility) and possibly taking into account practical considerations, such as the expected cost of the plan and its probability of success, chooses a plan to be executed by the execution method. While the elementary actions composing the plan are executed by components in the operative plan layer, for each non-elementary task $ACT_i$ included in the current active plan $P$ a new intender $I_i = \langle ACT_i, IC_i, P_i, IM \rangle$ is generated.

Since the environment where an intender operates is generally dynamically evolving in a non-deterministic way, the intender has to cope with several exceptional conditions. The truth values of the applicability conditions as well as their associated degree of credibility can change over time, or an intender generated to perform a non-elementary task may fail: in such cases, a revision method must determine whether to maintain the commitment to the current plan, select an alternative plan, or definitely withdraw the goal, notifying a failure to the generating mental entity. It may also be the case that the goal of the intender is satisfied independently of the actions it carries out: for this reason, an attender having $S$ as subject is always generated and, if it turns out that the desired state of the world has been reached, the execution of the plan $P$ is aborted.

In this context, a particular role is played by primary intenders: since they correspond to permanent goals defined at agent design time, they do not depend on the achievement of a specific goal, but are in charge of constantly monitoring external situations that may require their intervention. For this reason, primary intenders do not generate any attender associated to their subject, but they always generate an attender associated to the triggering condition $IC$ under which their intervention is required. Therefore, these primary attenders are always active and represent the basic attention mechanism of the agent.

Finally, it is also possible that different intenders try to access simultaneously a shared resource: this conflict is detected by the operative component managing the shared resource and must be solved by the involved intenders by means of their conflict resolution method. The mechanism for conflict resolution presented in the examples discussed in the next section is based on a simple comparison between priorities (Baroni & Fogli, 2000); we plan to revise this mechanism and to extend it with a sound argumentation-based approach in a future work.

### 3.5 Distributed argumentation and active mental entities

The active mental entities approach and the relevant architecture based on intenders and attenders were firstly introduced in (Baroni & Fogli, 2000) where ad hoc developed solutions were adopted to manage the interactions among active mental entities. While these solutions were applicable in specific contexts, they lacked generality and an adequate formal background.
The distributed argumentation algorithm presented in section 2 provides a more solid theoretical and computational foundation to our architectural proposal. In fact, it gives a basis to study the properties of the distributed reasoning mechanism constituted by attenders, and (in future work) to relate intenders activity with the results obtained by Pollock in the field of centralized argumentation-based planning.

As a technical detail, it should be noted that the distribution grain of the model assumed by the algorithm is finer than that of the active mental entities model. In particular, in the case of attenders, a single argument \( \alpha \) is not an independent process, but an element of the set \( \Sigma_T \cup \Sigma_F \) managed by the attender. As a consequence, all the arguments for a proposition \( S \) and its contrary \( \neg S \) are managed by a unique attender \( A = (S, V, \text{str}, \Sigma_T, \Sigma_F, \text{AM}) \), rather than by distinct processes. Of course, it is possible to conceive an approach where even the activity of each attender is distributed, for instance if an underlying massively parallel computing architecture makes it convenient. On the other hand, it is also possible to make the grain of distribution coarser, by allowing the attenders to internally construct whole parts of the arguments, rather than generating further attenders. The study of these alternative solutions, involving a trade-off between the potential of parallel computation and the additional communication overhead introduced, is beyond the scope of the present paper and reserved to future work.

In any case, it has to be remarked that our framework for distributed argumentation provides a uniform basis for these developments, applicable independently of the grain of distribution adopted and ensuring the correctness of the overall defeat status computation.

4. Examples

As mentioned in the introduction, we consider as reference scenario for the experimentation of our architecture the AAAI robot competition “Hors-d’oeuvre anyone?” (Gustafson, 2001), where the problem is to control a waiter robot which has to wander in a hall offering refreshments to the participants in a conference. A reasonable list of primary intenders for this kind of robot should include at least: Preserve-Robot-Integrity (RINTegrity), Preserve-Energy (ENERGY), Preserve-People-Integrity (PINTEGRITY), Offer-Drink-to-People (OFFERDRINK), Satisfy-Users-Requests (SATISFY), Go-Around (GOAROUND).

The intender RINTegrity represents, in a sense, the instinct of the robot to preserve its physical integrity: in our example, it basically exploits a plan to avoid obstacles. The triggering condition of RINTegrity is therefore the proposition “obstacle-in-front”, which can be verified through sensorial activity (e.g. sonar or visual) or by exploiting a map of the environment. The intender PINTEGRITY is similar to RINTegrity, but its activity is triggered
in the specific situation in which the obstacle is a person. Since we want to guarantee the harmlessness of the robot, we assume that the priority of PINTEGRITY is higher than that of any other primary intender. The intender ENERGY corresponds to the physiological need of maintaining the energy of the robot at an acceptable level; therefore, when the triggering condition “energy-under-minimum-threshold” becomes true, it executes a plan for battery recharge. The other primary intenders are related to the specific task the agent is in charge of, namely offering drinks to people: SATISFY is always active in order to respond to people which explicitly request the intervention of the robot (we suppose this is achieved by a speech recognition system), while OFFERDRINK makes the robot approach people, which might appreciate a drink even if they do not make an explicit request. Finally, GOAROUND pursues the goal of covering the entire working area assigned to the robot: it is enabled when the robot has stayed in a place for a while, and makes it reach another part of the hall, following for example a routine path.

4.1 Resolving conflict among attenders

In order to show how the role of argumentation comes into play, we consider two examples in which conflicts among attenders are involved. First, suppose that the primary attender with subject “energy-under-minimum-threshold”, which continuously monitors the energy level sensor of the robot, signals ENERGY that its triggering condition has become true. This enables a plan which involves the actions of going to the recharging area and of waiting for the completion of battery recharge.

Suppose now that, to go to the recharging area, the robot exploits a map which indicates the positions of different kinds of obstacles in the working area. These obstacles range from those which are definitely immobile, such as columns and walls, to movable obstacles, such as tables and chairs. Notice that there might be also objects, such as big flower vases, which are in the middle of this range. As a consequence, the information about obstacles given by the map have to be assigned different degrees of credibility, depending on the kind of object considered and, of course, on the reliability ascribed to the map. If, according to the map, there are no obstacles in front of the robot, the attender associated to the proposition “obstacle-in-front” is able to construct an argument supporting the negation of its subject. However, the strength of this argument is relatively weak, since, due to the presence of movable objects, the information provided by the map about empty places is not completely reliable. As a consequence, should a reliable sensor (such as the sonar) detect the presence of an obstacle, a stronger counter-argument supporting “obstacle-in-front” would win the conflict, changing the truth value of the subject and eventually enabling the intender RINTEGRITY. On the other hand, it might happen that the map indicates the
presence of a column in front of the robot, while the opposite indication is provided by the available sensors. In this case, however, the conflict is won by the argument based on the map knowledge source, which is considered definitely reliable as far as immobile objects are concerned (the sensor are therefore supposed to be malfunctioning); the RINTEGRITY intender is enabled and the robot can avoid the obstacle. This simple example shows the flexibility of the distributed argumentation mechanism: all information available can be exploited in determining the truth value of a proposition of interest, taking into account both the degree of reliability of different knowledge sources and the uncertainty affecting the defeasible rules of inference used in the construction of the arguments. Moreover, the dynamic generation of attenders, driven by the interests of the agent, provides a natural and effective focusing capability: the agent does not consider any possible piece of knowledge nor any data from any sensors to form a global coherent representation of the world. Instead, each attender only exploits those pieces of knowledge and those sensory data which are directly relevant to its particular task. Moreover, since the generation of active mental entities is driven by primary intenders, each attender represents a motivation for acquiring information about a given proposition. In this respect, two properties of the proposed model are remarkable. On one hand, the acquisition of data is goal-directed, always related to the motivations of the agent. On the other hand, the distributed and asynchronous nature of the model achieves an efficient information handling, where partial data are exploited to draw inferences and make decisions as soon as they are available, while additional information can always be considered in a subsequent moment, possibly leading to the revision of the decisions previously made.

4.2. A more complex deliberative behavior

In order to illustrate a more articulated reasoning activity supported by our approach, we consider an example involving the estimation of the age of a person which has requested a drink. We suppose that the intender OFFERDRINK makes use of two alternative plans, depending on the age of the person the robot is going to interact with: one plan has the applicability condition “¬ person-is-an-adult” and prescribes to offer a soft drink, while the other one has the applicability condition “person-is-an-adult” and allows for a wider choice including alcoholic drinks. In this case, the attender with subject “person-is-an-adult” tries to construct arguments for both applicability conditions. We suppose that the robot is endowed with a vision system, by which it can detect some particular features of objects (such as dimension and movement), and, in particular, exploiting an image processing algorithm, it is able to identify the conference badge of the participants. Moreover, the robot has an audio system which can be exploited to recognize and
analyze speech. The knowledge base of the agent is assumed to include the following defeasible rules of inference with the relevant strength factors:

- “object-moves ⇒ is-a-person” [s1]
- “object-speaks ⇒ is-a-person” [s2]
- “is-a-person ⇒ is-an-adult” [s3]
- “high-speech-frequency ⇒ ¬is-an-adult” [s4]
- “has-badge ⇒ is-an-adult” [s5]

Of course, we suppose s1<s2. The third rule expresses the default that a person should be a participant in the conference, and, as such, should be an adult; this is a very uncertain rule, therefore we have that s3<s4 and s3<s5. Moreover, it is reasonable to assume s4<s5, because the speech frequency does not definitely distinguish adults from children.

The operation of the attender in charge of assigning a suitable truth value to the proposition “person-is-an-adult” might proceed as follows. First, the attender constructs two arguments for “is-an-adult”, based on the premise “object-speaks” and “object-moves” respectively. According to the weakest link principle, the strength of both these arguments is equal to s3. Since “is-an-adult” becomes true, the robot starts the plan for adults. After a while, the audio system analyzes the speech frequency, which we suppose to be high, therefore the proposition “high-speech-frequency” becomes true and an argument for “¬is-an-adult” is constructed, which defeats the previous argument. However, if the video system subsequently detects the badge, the attender infers that the person must be an adult. Assuming that both visual and audio data are sufficiently reliable, the conflict is won by the latter argument, since s4<s5, and the first conclusion “is-an-adult” is reinstated.

5. Comparison with related works

In order to compare our approach with existing proposals, we consider three basic design dimensions of agent architectures, namely:

- **computation**, that concerns the logical organization of the computational entities, in terms of processes and threads;

- **action selection**, that concerns the mechanisms for selecting the actions to be carried out. Such mechanisms may range from hard-and-fast behavioral rules to more articulated deliberation mechanisms;

- **motivation**, that concerns the reasons underlying agent behavior, namely the origin and selection of the goals the agent is pursuing.

A clear trend in the evolution of agent architectures towards distribution, both in the computation and action selection dimensions, can be noted in the literature. In particular, pure behavior-based architectures (Brooks, 1986) put forward a distributed processing organization based on concurrent behav-
iors and a reactive action selection mechanism. As far as the motivation dimension is concerned, an implicit, hardwired representation of goals is adopted, an explicit symbolic representation of mental attitudes being rejected (Brooks, 1991). In order to combine reactive behaviors with explicit symbolic representations, various types of *hybrid architectures* have been proposed in the literature - see (Baroni & Fogli, 2000) for a review. In particular 2- and 3-layer architectures adopt a distributed computation organization and present hybrid action selection mechanisms, based both on planning and on reactive mechanisms. The motivation dimension is explicitly modeled through the symbolic representation of goals, beliefs, and intentions, which are considered as data structures on which computation mechanisms operate. In these approaches, distributed organization is one of the key aspects for achieving a flexible and extensible design. For instance, both Atlantis (Gat, 1992) and 3T (Wasson et al. 1998) are 3-layer architectures obtained by directly extending the 2-layer RAP system (Firby et al. 1995) with a deliberation level.

However, also in these works, distribution is limited to the dimensions of computation and action selection, while the motivational dimension is dealt with by managing a centralized database of mental attitudes. In fact, in these approaches the motivational dimension is added on top of other layers, rather than being really integrated with them, so that the advantages of a distributed organization are partially lost. For example, INTERRAP (Fischer et al., 1996) consists of three control layers, the *behavior-based layer*, the *local planning layer* and the *cooperative planning layer*, each one dealing with a specific level of abstraction and operating through a centralized engine on a knowledge base. There are two types of interaction among layers: *bottom-up activation* and *top-down execution*. In particular, bottom-up activation is the basic control flow that is started when perceptual input are received at the behavior-based level. When this layer is unable to manage the current situation, the control is passed to the local planning layer, without taking into explicit consideration, however, the relations between low-level behaviors and the underlying motivations. Moreover an agent can not organize its perceptual activity according to motivations, and to look autonomously for the information it needs.

On the contrary, our approach enforces a tighter integration of the motivational level with other ones. On one hand, motivation management may exploit mental entities generation and interaction in order to improve “reactivity”. For instance, intenders, which act autonomously without referring to a centralized goal/belief database, are able to generate attenders which monitor interesting aspects of the environment on their behalf, so that they can readily react to environment changes, thus exploiting favorable opportunities and avoiding risky situations. On the other hand, simple low-level behaviors, even basic reflexes, which in most agent architectures are exe-
cuted according to hardwired schemes, in our approach are associated with their motivation, so that any action carried out by the agent has an explicit justification. For instance, obstacle avoidance, which is in general appropriate, might be disregarded in case the agent accepts the risk of suffering damage for the sake of satisfying a specially important requirement (for example, assuring safety of human beings).

The full integration among the three design dimensions has been obtained in our approach by distributing not only computation and action selection, but also the motivation dimension. This requires a suitable theoretical framework to manage interactions among mental entities, which is provided by distributed argumentation. The adoption of argumentation theory is advantageous in many respects, some of which have been reviewed in Section 2, and further distinguishes our proposal from the aforementioned agent architectural approaches. For the sake of this comparison, literature approaches to argumentation can be roughly partitioned into three classes:

– specific frameworks of argumentation, or, at a more general level, of multi-context systems, without explicit reference to their integration in an agent architecture - see for example (Giunchiglia & Serafini, 1994; Elvang-Gøransson et al., 1993);

– argumentation-based monolithic architectures, where a single centralized reasoner controls the overall cognitive activity of an agent - see for example (Pollock, 1995);

– approaches where different agents adopt argumentation to negotiate, in order to coordinate their behavior and correct inconsistencies in their knowledge of other agents’ views - see for example (Parsons et al. 1998).

The adoption of a fully distributed agent architecture clearly distinguishes our proposal from the first two classes. As far as the third class is concerned, the relevant approaches focus on distribution of argumentation activity at the inter-agent level, while we deal with distribution at the level of internal cognitive activity of a single agent. For instance, the architecture proposed in (Parsons et al. 1998) includes three concurrent and interconnected units which manage beliefs, desires and intentions of an agent, and a communication unit which is responsible for enacting the agent’s communication needs. These units, which run in parallel, are implemented as centralized theorem provers, each one associated with a class of mental attitudes. Our approach allows a finer grain of distribution, and, as a consequence, the establishment of relationships between specific instances of active mental entities, rather than at the global level of the managers of mental attitudes, that is of theorem provers. As discussed above, this makes it possible to
obtain mechanisms for attention focusing and reactive behavior, which may be hardly achieved through interaction between theorem provers.

6. Conclusions

The key role of an articulated model of cognitive activity for the achievement of intelligence and autonomy in software agents has been advocated by several authors - see for example (Castelfranchi, 1995). This paper provides a contribution to this research direction by proposing the notion of active mental entity, i.e. by conceiving mental attitudes as distinct computational entities running concurrently. Argumentation has been identified as a suitable theoretical framework in order to ensure that the results of the distributed computation activity carried out by active mental entities obey a well founded semantics. To this purpose, a novel distributed argumentation algorithm has been developed. Two classes of active mental entities, namely intenders and attenders, have been introduced in order to substantiate the proposed paradigm. Among the main advantages of our approach we mention:

- a distributed organization of cognitive activity is coherent with the distributed paradigm of computation and action selection, adopted in several recent agent architectures; moreover this allows a deeper integration in the design of the different architectural dimensions of a software agent, including motivation;

- distribution of the cognitive activity among active objects makes the model open to extensions; the cognitive model of an agent can be progressively extended with a richer endowment of types of mental entities.

A prototypical implementation of the proposed architecture has been developed using “Actor Foundry”, an actor-based environment from the Open Systems Laboratory of the University of Illinois. Work is currently underway to carry out an experimental activity in a simulated robotics environment, in order to verify the advantages and the limitations of the approach.

References


