DiPRA: Distributed Practical Reasoning Architecture*

Giovanni Pezzulo, Gianguglielmo Calvi, Cristiano Castelfranchi
Istituto di Scienze e Tecnologie della Cognizione - CNR
Via San Martino della Battaglia, 44 - 00185 Roma, Italy
giovanni.pezzulo@istc.cnr.it; gianguglielmo.calvi@noze.it; cristiano.castelfranchi@istc.cnr.it

Abstract

DiPRA (Distributed Practical Reasoning Architecture) implements the main principles of *practical reasoning* via the distributed action selection paradigm. We introduce and motivate the underlying theoretical and computational peculiarities of DiPRA and we describe its components, also providing as a case study a guards-and-thieves task.

1 Introduction

Practical reasoning [Bratman *et al.*, 1988] is a kind of reasoning which is focused on the role of Intentions. BDI ("Belief, Desire, Intention") [Rao and Georgeff, 1995] is the most famous agent architecture implementing it, which underestimates however some architectural and cognitive features such as resource-boundedness, knowledge-boundedness and context-sensitiveness [Bratman *et al.*, 1988].

There are four main functions of practical reasoning: *means-ends reasoning*, *opportunity analysis*, *filtering* and *deliberation*. The peculiarity of practical reasoning is that these operations are managed in a plan-centered way: the adopted plan, filled in with the Intention, drives means-ends reasoning (plans are means for the end, the Intention), provides constraints for analyzing and filtering opportune options (only options which are relevant with the current intention are evaluated) and sets a priority level for its beliefs (only relevant beliefs will influence further practical reasoning). The rationale behind our work is that a rational agent architecture performing practical reasoning can be implemented as a *modular and parallel* system, in which each Belief, Goal, Action and Plan is a module operating asynchronously (with different activity levels) and having relations with other modules (such as: *Belief* β *supports Goal* γ). A special module, *the Reasoner*, maintains a consistent representation of the modules' activation level and their relations by using a Fuzzy Cognitive Map (FCM) [Kosko, 1986]. It weighs the alternative goals (exploiting a mixture of means-ends reasoning, opportunity analysis and filtering) and deliberates. There is a continuous interplay between the Reasoner and the other modules: after selecting a (new) Intention the Reasoner assigns to modules an activity level (i.e. the thread's priority) proportional to their values in the FCM (and thus, as we will see, to their contextual relevance), so that more relevant modules influence more the computation. At the same time, the modules act in the environment and provide feedback for the values of the FCM used by the Reasoner. For example, a Condition can be verified or falsified by an action of the agent (in the example we will provide, detecting if a door is open or close), or a Plan can succeed or fail; the results are notified to the Reasoner which updates the values of the corresponding nodes in the FCM. As a result, practical reasoning is realized with central deliberation and a decentralized control structure: differently from BDI Interpreters, the Reasoner simply activates the (modules encapsulating) adopted plans, but after this phase the control flows between the modules in a dynamic way. Plans activate actions and subgoals without a new intervention of the Reasoner; any further deliberation (choosing subgoals) is performed inside the plan. The activity of the modules (success of action, testing of beliefs and conditions) provides feedback to the Reasoner, too.

In this work we only focus on *present-directed Intentions* [Bratman *et al.*, 1988]: Intentions which are selected to be activated here and now. We illustrate DiPRA, a modular, parallel and resources-bounded architecture, arguing that it permits to model the four functions of practical reasoning as an interplay of knowledge, goals, contextual factors and opportunities; we also provide a case study.

2 DiPRA Specification and Components

The components of DiPRA are: *the Reasoner, Goals, Plans, Actions and Beliefs*. Each of these components is implemented as a concurrent *module* in the multi-thread framework AKIRA [akira, 2003]; DiPRA is also interfaced with an environment (e.g. the physical simulator [irrlicht, 2003]), called the *World Engine*, which evaluates its actions.

Let S be a set of worldstates, P^+ a set of atoms, and $\pi : P^+ \times S \rightarrow [0..1]$ a function assigning a truth value to each atom in each worldstate. P is a set of atoms and negated atoms where $\pi(p, s) == 1 - \pi(\neg p, s)$. L is a propositional language over P and the logical connectives $\wedge$ and $\vee$, where: $\pi(p \wedge q, s) := \pi(p, s) \otimes \pi(q, s)$ and $\otimes$ is any continuous triangular norm (e.g. $min(p, q), pq$); $\pi(p \vee q, s) := \pi(p, s) \oplus \pi(q, s)$ and $\oplus$ is any continuous triangular conorm (e.g. $max(p, q), x + y - xy$) (see [Saffiotti *et al.*, 1995]).

DiPRA is described by a tuple $(\Psi, \Gamma, \Pi, \Phi, Bel, \Omega)$, where:

*Work supported by the EU project **MindRACES**, FP6-511931.

- Ψ is the *reasoner*, a tuple (*FCM, Body*). **FCM** is a Fuzzy Cognitive Map [Kosko, 1986], a representation of the state and the relations between all the modules; **Body** is the procedural body, whose main task is to assign the status *intended* to a goal and *adopted* to a plan.

- Γ is the set of *goals*, tuples (*Type, Status, GCond, AbsRel, ConRel*). **Type** is the type of goal: *Achieve* or *Maintain*; **Status** is the current status of the goal: *Intended*, *Instrumental*, *Waiting* or *Not Intended*; **GCond** $\in L$ is the (graded) satisfaction condition of the goal; **AbsRel** is the absolute relevance; **ConRel** is the contextual relevance.

- Π is the set of *plans*, tuples (*Status, SCond, ECond, PCond, ActionSet, Body, Goals, Results, AbsRel, PCondRel, ConRel*). **Status** is the current status of the plan: *Adopted* or *Not Adopted*; **SCond** is the set of start conditions $sc \in L$ which are checked at the beginning of plan execution and must be true to start it; **ECond** is the set of enduring conditions ec $\in L$ which are checked continuously during plan execution; if an enduring condition becomes false, the plan is stopped; **PCond** is the set of beliefs $\beta \in L$ which are expected to be true after the plan (but not all of them have to be intended); **ActionSet** is the set of actions ϕ or (sub)goal γ activated by the plan. Actions and goals are chained inside ActionSet by logical connectives λ; **Body** is the behavior which is executed; it normally consists in activating actions and (sub)goals in the ActionSet; **Goals** is the set of goals γ that make the plan satisfied; they are the subset of PCond which are intended (the reasons for activating the plan); **Results** is the set of the final plan results $pr \in L$, corresponding to the GCond of the Goals at the end of the plan; **AbsRel** is the absolute reliability value of the Plan, i.e. how reliably it succeeds; **PCondRel** is the set of the reliability values $ar \in L$ of the plan with respect to its PConds, i.e. how reliably it produces its PConds; **ConRel** is the contextual relevance.

- Φ is the set of *actions*, tuples (*SCond, PCond, Body, Goals, Results, AbsRel, PCondRel, ConRel*). **SCond** is the set of start conditions $sc \in L$ which are checked at the beginning of action execution and must be true to start it; **PCond** is the set of beliefs $\beta \in L$ which are expected to be true after the action (but not all of them have to be intended); **Body** is the behavior which is executed once the action is executed; **Goals** is the set of goals γ that make the action satisfied; they are the subset of PCond which are intended (actually the reasons for activating the action); **Results** is the set of the final action results $ar \in L$, corresponding to the GCond of the Goals at the end of the action; **AbsRel** is the absolute reliability value of the Action, i.e. how reliably it succeeds; **PCondRel** is the set of the reliability values $ar \in L$ of the action with respect to its PConds, i.e. how reliably it produces its PConds; **ConRel** is the contextual relevance.

- Bel is the set of epistemic states, i.e. beliefs $\beta \in L$. All the conditions (GCond, SCond, PCond, ECond) are kinds of beliefs. Bel are tuples (*β, AbsRel, ConRel*). $\beta \in L$ is the value of the belief or condition; **AbsRel** is the absolute relevance; **ConRel** is the contextual relevance.

- Ω is a set of *parameters* used to control the energetic dynamics of the modules: σ is the activation of the reasoner; ζ is the threshold for goal intention; η is the threshold for plan adoption; θ is the commitment level of intended goals; κ is the commitment level of adopted plans; ϑ is the total amount of resources available to the whole system.

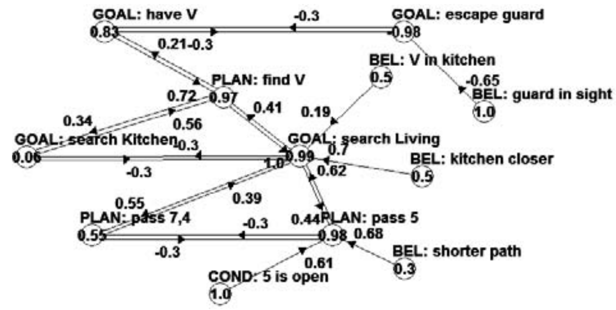

Figure 1: The FCM used for the Thief in the House Scenario

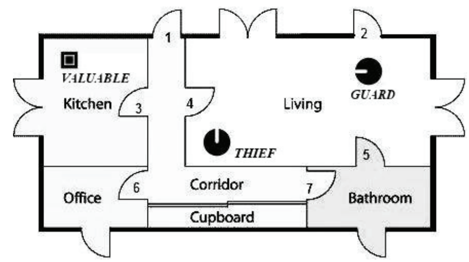

Figure 2: The House Scenario (description in the text)

2.1 The Reasoner

The Reasoner maintains a consistent representation of the activity of all the modules and performs deliberation using an additive fuzzy system called FCM [Kosko, 1986] whose nodes and edges represent the modules and their links, and in which activation spreads between nodes. Fig. 1 shows a sample FCM in the House Scenario (Fig. 2, introduced later).

Deliberation consists in intending a goal and adopting (the best) plan for it; it is performed by the FCM by weighting the alternative plans and goals and, at the same time, by evaluating chains of goals and plans, including of course conditions. In traditional practical reasoning there are three different mechanisms for generating alternatives: means-ends analysis, opportunity filtering, filter overriding. The FCM formalism permits to represent the constraints of all these mechanisms in a compact way, and to provide at the same time suitable values for deliberation. The FCM can represent many typical situations in practical reasoning: Goals concurrence (via inhibitory links); Beliefs sustaining a Plan or a Goal; a Plan preferred to another one because one of its preconditions is already matched; a Goal activating one or more Plans which are able to satisfy it, etc.

In the FCM there are six kinds of (weighted) links: (1) *Satisfaction* A goal links plans and actions whose PCond satisfy its GCond; in this way, activation is spread from intended goals to plans which realize them. (2) *Predecessor* A plan links (sub)goals whose GCond realize its SCond or ECond; in this way, if a plan has a missing PCond or ECond it can "subgoal". (3) *Support* A belief links goals, plans or actions which correspond to their GCond, SCond or ECond; this is a way to represent contextual conditions: goals, plans and actions which are "well attuned" with the context are preferred. (4) *Feedback* a plan or action feedbacks on goals; this special case of support permits to select goals having good paths to action (5) *Inhibition* Goals and plans with conflicting GConds or PConds, and plans realizing the same GCond have inhibitory links. In this way it is possible for a goal or plan (especially if intended or adopted) to inhibit competitors. (6) *Contrast* Beliefs have inhibitory links with plans and goals having conflicting GConds, PConds, SConds and EConds: this is a kind of "reality check".

The Cycle of the Reasoner

The Reasoner (and the FCM) runs concurrently with all other modules, having an activation σ: in a real-time system, even reasoning takes resources. The Reasoner has two main tasks: (1) to deliberate (select a goal and a plan) and (2) to set the activation of the modules. Both are realized in this cycle:

1. Set the values of the FCM nodes according to the activity level of the corresponding modules[1], and their links;

2. Run the FCM and obtain the values of the nodes; as explained later, this value represents the *contextual relevance* (ConRel) of the corresponding modules;

3. The most active Goal is selected (if over a threshold ζ); if not already achieved, its status becomes *Intended* (intended Goals replace old intended ones). Otherwise, another Goal has to be selected. A recurrent connection (with weight θ) is set for the Intended Goal, which thus gains activation[2].

4. The most active Plan for the intended Goal is selected (if over a threshold η); its status becomes: *Adopted*. The plan is filled in with the intended goal: the Goal of the plan becomes the GCond. If there is an already adopted Plan, it is stopped only if its PCond conflict with the conditions of the new adopted one. A recurrent connection (with weight κ) is set for the Adopted Plan.

5. If no Plans are possible for the intended Goal, its status becomes *Waiting* (and maintains the recurrent connection); a new Goal has to be intended (this is unlikely, since the evaluation of a Goal also depends on how suitable are its plans);

6. If no goals or plans are over the thresholds ζ and η, the thresholds lower and the cycle restarts. Otherwise sets the activity level of the modules to the value of the nodes in the FCM. Thus, even if the Reasoner runs concurrently with the other modules, it resets their activity level only if a new Plan is adopted (either or not a new Goal is intended). *Resources boundedness* is guaranteed by the parameter ϑ: the total amount of activation to be assigned to all the modules can be fixed so that the computation never exceeds that threshold.

[1]Goals, plans, actions and beliefs can also be *more or less relevant in absolute*; this is represented by the AbsRel value, which is also the value of a recurrent connection in the corresponding node in the FCM (not shown in Fig. 1). As a result, the activation and influence of more relevant/reliable modules grow faster than others.

[2][Castelfranchi and Paglieri, 2007] argues that some characteristic supporting beliefs are also necessary for Intending a goal. Here we do not check them and simply assume that it is always the case.

Contextual Relevance and Impact. The value of the nodes in the FCM represent the *contextual relevance* of the corresponding modules. The value of the edges in the FCM represent the *impact* of the corresponding modules; by default they are set according to the "epistemic component" of the module: the value of a Belief, the GCond of the goal, the SCond or ECond of the plan. For example, if a belief has $\beta = 0.4$ and its sustains a goal, the impact of its edge in the FCM is $+0.4$. The impact of the modules varies during the computation; for example, an achievement goal which is close to satisfaction inhibits more and more its competitors.

Not all the modules have to be represented at once in the FCM (and not all the threads have to run). FCM nodes having contextual relevance equal (or close) to zero have no impact and can be deleted (and the threads of the corresponding modules stopped): in this way only relevant knowledge is considered, and the FCM never exceeds a certain size. This feature is very useful in means-ends analysis: at the beginning, only top-level plans are considered in the FCM; plans (and the FCM) are filled in with subplans only as long as the activity proceeds. Knowledge augments in a bounded way, too, as long as conditions and Beliefs related to active Plans, Actions and Goals are checked.

2.2 Beliefs, Conditions and Goals

All the declarative components use fuzzy logic [Kosko, 1986]. All the conditions (goal conditions, pre and post conditions of plans and actions, etc.) are special kinds of beliefs. For example, a Belief ("Office is far") can be matched using fuzzy rules with the PreCondition of a Plan ("Office is close") and generate a graded truth value. Also goal conditions share this formalism; in this way they can be matched e.g. against post conditions in order to verify their satisfaction (e.g. the Goal "go to Office" becomes more and more satisfied when the truth value of "Office is close" increases).

There are policies for both *achieve* and *maintain* goals. In Achievement goals (such as "reach Office"), the contextual relevance increases on nearing the goal (when the truth value of the GCond increases). In Maintain goals (such as "stay close to Office"), the contextual relevance lowers on nearing (when the truth value of the GCond increases).

Intended vs. Instrumental Goals. In practical reasoning it is assumed that only one goal is Intended, but many goals can be active at once (and activate plans or actions); they are named *instrumental* goals as opposed to *intended* ones; their purpose is to favor the intention, e.g. by creating appropriate contextual conditions. If the intended goal (or another instrumental goal) they depend on is achieved, they are stopped.

2.3 Plans

Plans are the main control structures in DiPRA; they do not depend on the Reasoner except for starting. Plans are activated for satisfying an intended goal; once the plan is adopted, a subset of their PCond is set as Goal. A Plan is basically an execution scheme, activating Actions and Goals from the ActionSet and subgoaling; this is their behavior:

- If the intended Goal is already achieved, the Plan returns immediately and no action is executed.

- If any SCond or ECond is false, the Plan "delegates" their satisfaction to other modules by passing them activation via the *Predecessor* links; subgoals activated in this way gain the status of *Instrumental*.

- If all the SCond and ECond are met, the Plan starts executing the actions in the ActionSet, chaining them according to the connectives in the ActionSet[3]. Plans can load from the ActionSet not only actions, but even goals. This mechanism produces the typical subgoaling of practical reasoning: (sub)goals activate (sub)plans or actions, and so on. Even goals activated in this way gain the status of *Instrumental*.

- Plans continue subgoaling and executing their body until all the possible actions and subgoals fail. A failed plan returns the control to the calling goal, which remains not satisfied and activates another plan. However, it is likely that unsuccessful plans are stopped before exhausting all the possibilities; in fact, if many conditions of a plan fail, despite commitment it weakens in the FCM and other plans replace it.

Plans and Subgoaling. There are two subgoaling mechanisms realized by the plans: the first one consists in activating goals which realize their SCond and ECond (if they are not already realized); the second one consists in activating goals instead of actions from the ActionSet. Both kinds of goals are *Instrumental*. At the same time, plans spread activation to instrumental goals, which gain priority.

2.4 Actions

An Action is the minimal executable operation; typically it consists in an interaction with the *World Engine*; but actions can also check, add, remove or modify a Belief (*epistemic actions*). Actions are activated by goals whose GCond correspond to their PCond or by Plans via the ActionSet.

Post Conditions vs. Goals. In practical reasoning, not all the expected results of actions and plans are intended. When the plan is adopted, one of its PCond (corresponding to the intended goal) is selected and becomes the Goal; the same happens to actions. Depending on the situation, actions and plans can be activated for different reasons: their post conditions are the same, but the Goal is different.

2.5 Dynamics in DiPRA

Even if there is only one intention, many modules can be active at once in DiPRA. Not intended and not adopted goals and plans have a certain amount of activation, too, which can be used for fulfilling operations such as building up parts of the FCM, although these operations tend to be slower.

Goal-Driven Pressures. Goals represent desired states of the system. In DiPRA an active goal "drives" the computation toward a certain result (such as "Office") in three ways: (1) By actively competing for being intended: in this way they can lead to adopt an explicit plan leading to Office. (2) By indirectly introducing a pressure. Even not intended goals have an influence which is proportional to their activation. For example, in choosing between two plans, an active but not intended goal can do the difference (e.g. by reinforcing a plan whose PCond are close to its GCond, or by weakening a plan whose PCond are far from its GCond). (3) By updating knowledge related to them (e.g. GCond); in this way more beliefs which are pertinent to the goal (and in principle can reinforce it or weaken the other ones) are produced.

[3]Actions are set independently on any control structure such as Plans, that only order them. Depending on the connectives the same set of actions can be executed in different ways. For example, the *OR* connective can be used for running two actions in parallel.

Epistemic Dynamics. Knowledge is distributed and available to different extent to deliberation, depending on the activation level of the modules corresponding to beliefs. For example, not all the consequences of adopting a plan (e.g. subplans, PConds) can be considered in means-ends analysis, but only those currently available; this is why sometimes long term conflicts are discovered only after a plan is adopted. This is represented by putting in the FCM only beliefs, plans and goals having a non-zero contextual relevance.

It is important to note that more active Beliefs intervene more into the computation: they activate more the Goals and Plans they support. This aspect models their *availability*: for example, an highly active belief is ready to be exploited for reasoning and, if it *sustains* a goal, gives it more activation.

Beliefs are retrieved in an activity-based and bounded way: not all the knowledge is ready to be used, but modules actively search for and produce knowledge (and that activity takes time) with a bias toward knowledge useful in the context of the most active goals. Goals, plans and actions assign an updated truth value to their conditions (and to related beliefs) during their execution, for example by reading a sensor or asking memory; more active beliefs (receiving activation from more active goals and plans) will auto-update their truth value more frequently. Produced (or updated) beliefs are added to the current state (and to the FCM) and linked to the relevant goals, plans or actions: new knowledge can lead to intention reconsideration or to replanning. More active goals and plans can build longer means-ends chains and have more "up-to-date" knowledge, since they can perform more epistemic actions. The epistemic component of DiPRA (what the agent knows) is influenced by its current activity (what the agent is doing): in practical reasoning a crucial role of Intentions is selecting relevant information.

3 Practical Reasoning in DiPRA

In DiPRA *means-ends analysis*, *opportunity filtering* and *filter overriding* are "weak constraints" of the same mechanism and, at the same time, provide suitable values to deliberation.

Means-Ends Analysis and Deliberation. Means-ends analysis builds causal chains: what is necessary for achieving a goal. Deliberation evaluates utility: what is better for achieving a goal. These two activities are related. Means-ends analysis consists in building causal chains of *means* (of plans, (sub)goals, conditions and actions) to achieve *ends*. Normally this process is incremental: even if declarative knowledge about plans and their effects is already available, in order to fulfill new goals a "chain of means" has to be built anew. We have seen that the FCM is more and more filled in with elements of this chain, as long as the analysis proceeds (new nodes are added as the result of *epistemic actions* of the modules, e.g. a plan verifying its preconditions); and as long as the agent acts (its actions have consequences which can be added as beliefs). The rationale is that knowledge related to the goals and plans (e.g. about conditions and actions) becomes more and more "relevant" and is thus added to the FCM. Normally means-ends analysis is performed only for top-level plans, which are not totally filled in. However, plans whose chains (from top-level plans to terminal actions) are stronger (having reliable subplans and actions and true conditions) are privileged, because top-level plans gain more activation from them. As long as the analysis proceed, with or without adopting a plan (e.g. if the threshold η is not reached, or if there is another adopted plan), new knowledge about the plan is added and it can make it more likely to be selected.

Means-ends analysis, which is mainly qualitative, produces at the same time results which are suitable for deliberation, because the utility of a course of actions depends also on the availability of the conditions and the reliability of the actions. Since in the FCM the plan receives activation from all its conditions, while performing means-ends analysis the "best" plans receive also more and more activation. It is also very likely that the most active plan has many PCond and ECond already met (at least partially). In a similar way, plans having highly reliable actions are more likely to be very active. In this way, deliberation exploits the results of means-ends analysis: the values of the nodes in the FCM, built during means-ends analysis, can be directly used for selection (we provide as a simple heuristic: choose the highest one, but more sophisticated ones are possible).

All the preference factors normally related to Goals and Plans in the BDI (e.g. urgency, utility) are encoded into modules activation. Preference is mainly based on epistemic factors: Goals and Plans are activated by knowledge, that can be explicitly represented (e.g. "Goal_x is very important"), implicitly represented into the modules (e.g. a Pre Condition of a Plan) or encoded in the relations between the components (e.g. a link between a Goal and a Plan means that the Plan is able to satisfy the Goal). The rationale is that the belief structure of an Agent motivates its choices and preferences; the causal structure built by means-ends reasoning is also used for deliberation. There are two main difference with practical reasoning as traditionally implemented (e.g. in BDI): (1) conditions satisfaction and action evaluation are treated as "weak constraints"; (2) there is an active and bounded view of how knowledge to be evaluated is added.

Opportunity Analysis and Filtering. In traditional implementations of practical reasoning the consistency of new plans or goals with old ones is routinely checked; inopportune plans and goals are ruled out. Eventually, an intention which is discarded because of its incompatibility can be reconsidered in another mechanism, the filter overriding. In DiPRA these brittle and costly operations are replaced by "weak constraints" in the FCM: plans or actions which PCond conflict with existing states (or desired ones such as goals) are simply much less likely to be selected and, at the same time, become less and less relevant. This is mainly due to the *inhibition* and *contrast* links, but also to the fact that selected goals and plans create areas of high "relevance" around them: conditions and beliefs which potentially activate them are very likely to be added to the FCM.

In general, some requisites of practical reasoning (such as opportunity analysis) are perhaps too strong; we argue that a cognitive agent (with limited rationality and bounded resources) implements weaker requirements. For example, an intended goal or an adopted plan do not rule out their competitors, but simply gain more contextual relevance and weakens the other alternatives, too. New goals can be intended and new plans adopted if they are able to overwhelm the "weight" of the previous ones. Intention reconsideration (changing Goal) or replanning (changing Plan) only occur when needed. Once a Goal is intended, it only has to be replaced if a goal which is more important or was previously intended but was not executable becomes achievable. Once a Plan is adopted, it only has to be replaced if one of its ECond become false or if its Goal is no more intended. All these situations happen naturally in the FCM.

Commitment. The most distinctive point of a practical reasoning agent is that it is *committed* to its intentions (and to doing what it plans). Commitment, however, comes in grades, since agents should also be able to be opportunistic and revise their intentions. Commitment is implemented in BDI as a strict rule; in DiPRA is comes in grades and it is regulated by two parameters, θ and γ. Commitment to a Goal or a Plan is also maintained by the structure of the links, since achievement goals which are close to satisfaction impact more and more, and adopted plans are more and more reinforced by their conditions which increase their truth value.

4 A Case Study: The House Scenario

We implemented the House Scenario (see Fig. 2) using the framework AKIRA [akira, 2003] and the 3D engine [irrlicht, 2003]; the House has five rooms and seven doors which open and close randomly. The agent we model is the **Thief**; it appears in a random position in the house, having the achievement goal to possess the valuable **V** (that is hidden in the house) and the maintenance goal to avoid the **Guard** (an agent which moves randomly, but when spots the thief moves straight toward it). The Guard and the Thief have the same size and speed, and a limited range of vision. Four implementations of the Thief were tested: (1) *DiPRA*; (2) a *baseline* (random system); (3) the *A* algorithm* [Hart *et al.*, 1968] (which has full knowledge of the environment, including the location of V, plans the shortest path to it but and replans when something changes in the environment, e.g. a door closes); (4) a classic *BDI*, based on [Rao and Georgeff, 1995] (having the same goals, plans and beliefs of DiPRA).

Percentage of success (having V without being captured by the Guard) was measured in 100 runs: analysis of variance (ANOVA) shows that **DiPRA** (81%) performs significantly better than the other strategies ($p < 0,0001$ in all cases):

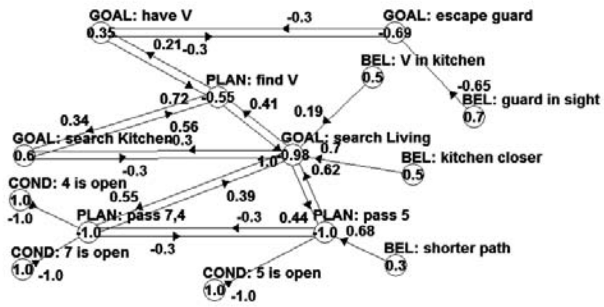

Figure 3: FCM after intention change (description in the text)

Baseline (12%), **A*** (56%) and **BDI** (43%). Resources and knowledge boundedness make DiPRA much more efficient in real time and dynamic situations.

Some situations occurred during the simulations may help illustrating the behavior of DiPRA. Consider the following case: the Thief is in the Bathroom and has the goal to find the valuable V, assuming by default that all the doors are open, and to escape the Guard. Fig. 1 shows the corresponding FCM, including two competing goals, *have V* and *escape guard* (note that all the horizontal links are inhibitory). Since only the former is contextually relevant (0.83 vs. -0.98), only its "means-ends" causal chain is constructed by DiPRA and included in the FCM (all the other goals, plans and beliefs are supposed to have a value close to zero). Given this contextual situation, with many possible goals and plans supported by many beliefs, the Thief *intends* the goal with the highest value (0.99), *search Living*; this goal is selected both because the living is the closest room and because the Thief believes that V is there. Now the Thief *adopts* the best plan (0.98) realizing the intended goal, *pass 5*. Actions (such as moves) are not shown in the FCM. Now, if door 5 is found closed, the Intention remains the same and a new plan is selected: *passing doors 7 and 4* (this situation is not shown here). If door 4 is close, too, it is impossible to realize any plan for the given Intention. Assuming that no subgoal (such as: *open door 4 or 7*) is possible, it is necessary to have a new Intention (e.g. *search the Kitchen*): the resulting FCM is shown in Fig. 3.

Another case of Intention reconsideration, different from plans failure, is a conflict between an Intention and another goal which becomes contextually active, i.e. an opportunity. For example, while the Thief has the Intention to search the Living (processing the plan to pass doors 4 and 7), it could spot the Guard near door 1. At that point, the goal to avoid the Guard comes in play, too, and it could be so strong to defeat the current Intention, becoming the new Intention.

5 Conclusions

Deliberation is implemented in BDI via a central interpreter, which selects goals and plans, updates knowledge and monitors the environment. DiPRA instead distributes control among semi-independent, parallel modules (goals, plans and actions) which are assigned an activity level proportional to their contextual relevance; more relevant goals and plans are more likely to be selected. Also knowledge is distributed among modules representing beliefs, preconditions or postconditions. The only central component, the Reasoner, is only responsible for setting the activity level of the modules once a new Intention is selected. *In DiPRA, practical reasoning is en emergent property of the modular architecture*.

An advantage of distributed systems is that many situations, such as the conflicts between Goals and means-ends analysis, are resolved on-line by a dynamic, anytime system. Many interesting dynamics emerge; for example, two conflicting Goals can influence one another even via energy dynamics in a way that varies with time. Or, a given Plan can start with many resources when a Goal is very powerful, be weakened when the Goal weakens, and be stopped when a conflicting Goal grows and inhibits the former. All these possibilities have not to be pre-planned, i.e. the exact moment when the Plan stops is not explicitly set but it depends on the dynamics of the system. Moreover, the relations of conflict or cooperation between two Goals have not to be always explicitly represented (with inhibition links) but can emerge as set points of the system's dynamics.

DiPRA is influenced by its expectations and monitors them. Goals represent desired and expected future states; Plans and Actions have explicit PConds. By activating Goals, Plans and Actions some "beliefs about the future" appear in the FCM and influence the deliberation. As it happens for all the beliefs, modules for testing PConds become active, too.

In our experiments the model has shown to be effective and scalable: the competition between Goals and Plans is credible; since only relevant modules are considered, even adding more goals, plans and actions the size of the FCM remains bound. The system is committed to its current Goals and Plans and it is smooth in shifting from one another. As shown in [Kosko, 1986], machine learning techniques such as *hebbian learning* can be used for learning the FCM, too.

References

[akira, 2003] akira, 2003. http://www.akira-project.org/.

[Bratman *et al.*, 1988] M. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[Castelfranchi and Paglieri, 2007] C. Castelfranchi and F. Paglieri. The role of beliefs in goal dynamics: Prolegomena to a constructive theory of intentions. *Synthese*, to appear, 2007.

[Hart *et al.*, 1968] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[irrlicht, 2003] irrlicht, 2003. http://irrlicht.sourceforge.net/.

[Kosko, 1986] B. Kosko. Fuzzy cognitive maps. *International Journal Man-Machine Studies*, 24:65–75, 1986.

[Rao and Georgeff, 1995] A. S. Rao and M. P. Georgeff. BDI-agents: from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, 1995.

[Saffiotti *et al.*, 1995] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artif. Intell.*, 76(1-2):481–526, 1995.