ARGUING OVER MOTIVATIONS WITHIN THE V3A-ARCHITECTURE FOR SELF-ADAPTATION

Maxime Morge[1], Kostas Stathis[2] and Laurent Vercouter[3]

[1]*Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3, Italy*

[2]*Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, U.K.*

[3]*Centre G2I, Ecole des Mines de Saint Etienne, 158, cours Fauriel, F-42023 Saint-Etienne, France*

Keywords: Multiagent systems, Agents models of motivation and personality, Agent architecture, Argumentation, Self-adaptation.

Abstract: The Vowel Agent Argumentation Architecture (V3A) is an abstract model by means of which an autonomous agent argues with itself to manage its motivations and arbitrate its possible internal conflicts. We propose an argumentation technique which specifies the internal dialectical process and a dialogue-game amongst internal components which can dynamically join/leave the game, thus having the potential to support the development of self-adaptive agents. We exemplify this dialectical representation of the V3A model with a scenario, whereby components of the agent's mind called *facets* can be automatically downloaded to argue an agent's motivation.

1 INTRODUCTION

Component-based engineering is a promising approach to develop adaptive systems. Adaptation is obtained by replacing some components by others or just by changing the connections between components. This approach has already been adopted to build agents (Ricordel and Demazeau, 2000; Meurisse and Briot, 2001; Vercouter, 2004) to ease the modification of the internal structure of an agent. A current challenge is to automate this process in order to provide self-adaptive agents. During this process, the coherence of an assembly of components must be warranted, i.e when the assembly is changed (addition/removal of components and/or connections) the result must be coherent. Solutions have been proposed to deal with these issues but are not completely satisfactory. The work of (Meurisse and Briot, 2001) proposes to foresee all the possible assemblies and seeks to describe what should be done in each case. This approach reduces the openness of the system only to foreseen situations. Other works rely on human intervention (Ricordel and Demazeau, 2000) or they do not warranty the coherence of the resulting behaviour (Vercouter, 2004).

In this paper we propose the Vowel Agent Argumentation Architecture (V3A) to design self-adaptive agents. Inspired by the Vowels approach (Demazeau, 1995), V3A is built upon an intentional stance. Additionally, V3A is divided in components called *facets* encapsulating the different motivations of an agent. We view these motivations as arguments describing the conditional decisions to achieve goals. A *facet* can join the internal debate to argue for/against the adoption of a motivation. Then, the personality arbitrates the possible conflicts. Regarding self-adaptation, this personality corresponds to high-level guidelines to solve conflicts appearing when modifying the component assembly. The contribution of the work is an abstract model of agency and the definition of a dialogue-game that can be played by facets which can dynamically join or leave the game. For this purpose, we consider an argumentation framework (Dung, 1995) built to realize this internal dialectical process within this modular architecture. A scenario illustrates the use of this mechanism.

The paper is organised as follows. Section 2 introduce the walk-through example to motivate our proposal. Section 3 presents the V3A model, Section 4 presents the argumentation framework used to support it, and Section 5 describes the dialogue-game facets play. We conclude in Section 6 where we also present related work and we discuss our plans for the future.

Morge M., Stathis K. and Vercouter L. (2009).
ARGUING OVER MOTIVATIONS WITHIN THE V3A-ARCHITECTURE FOR SELF-ADAPTATION.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 214-219
DOI: 10.5220/0001657802140219

2 WALK-THROUGH EXAMPLE

We motivate our approach with the following scenario. Max, who lives in Pisa, must reach London for a meeting. Since he does not like to spend too much time in queues in order to buy tickets, he has a PDA equipped with a personal service agent (PSA) in order to automatically buy a ticket when Max is in a train station. The PSA has been set up by the constructor to respect the user's preferences and to use the computing system in order to assist the user.

Our starting point is the Vowels approach of multiagent oriented programming (Demazeau, 1995), which has shown to be suited for developing agent platforms e.g. see (Ricordel and Demazeau, 2000) and (Vercouter, 2004). In this approach a multiagent system (MAS) is analysed across five dimensions: *Users*, *Agents*, *Environments*, *Interactions*, and *Organizations*. The Vowels approach is generally used in the analysis stage of the building process in order to divide the problem according to these dimensions.

Users. Max has configured his PSA for this travel and does not allow his PSA for paying extra commissions since he will not be refunded for these. When Max arrives in a railway station, the PSA automatically downloads the four following components.

Agents. The PSA has a representation of the available seller agent (SA).

Environments. The PSA have the knowledge about the current location, the current time, the possible traffic troubles, and the train time schedule.

Interactions. The PSA has a representation of the protocol required for communicating with the SA.

Organizations. The PSA has a representation of the norms adopted by the system for the automatic payment.

In this way, when Max arrives in the railway station of Pisa, the PSA is able to request a train ticket to reach Pisa airport. Since the train is not fully booked the SA sells one to the PSA and the payment is performed within the computing system. Therefore, Max has a reservation and can show its PDA to the controller agent on board. When Max arrives in Stansted, the PSA automatically replaces the four previous components. Since the SA overcharges the train ticket price with extra fees, the PSA does not register the user. Therefore, Max will pay the train ticket on board to the controller agent.

3 THE V3A MODEL

Our agent architecture (Fig. 1) consists of: a *KBase* (KB) partitioned according to the vowel dimensions ($\mathtt{KB_{user}}$, $\mathtt{KB_{agt}}$, $\mathtt{KB_{env}}$, $\mathtt{KB_{int}}$, $\mathtt{KB_{org}}$), an *Argumentation State* ($\mathtt{AS}$) and a *personality* ($\mathtt{pers}$). The KBase is a repository of rules and assumptions possibly conflicting. The parameter related to a dimension can still be too large and needs to be decomposed in different *facets* ($\mathtt{f}_1, \ldots, \mathtt{f}_7$). We partition further the dimensions linked to the external world to distinguish perceptions and capabilities. For instance, $\mathtt{KB_{env}}$ contains the representation of the physical laws in the environment ($\mathtt{KB_{env_1}}$) and the observations/actions perceived/performed by the body ($\mathtt{KB_{env_2}}$). Similarly, $\mathtt{KB_{int}}$ contains the protocols ($\mathtt{KB_{int_1}}$) and the messages received/sent by the agent ($\mathtt{KB_{int_2}}$). Facets joint (respectively leave) the game for adding (respectively deleting) particular aspects of the whole possible agenda of the agent, which taken together will comprise every aspects that the agent can address.

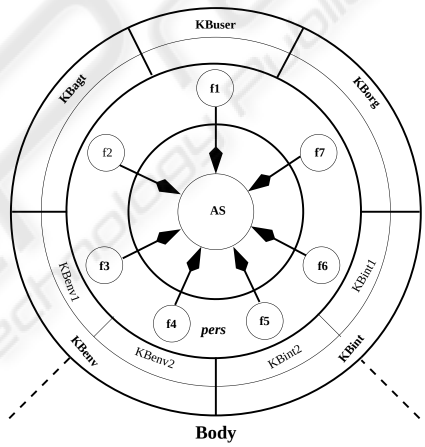

Figure 1: The V3A agent architecture.

The interaction between facets is organized as an argumentation game, consisting of dialogical moves about statements. The dialogue is regulated by procedural rules. $\mathtt{AS}$ is a shared memory structure. Its current state includes a (partial) argument not yet defeated or subsumed. At the end of the game, $\mathtt{AS}$ contains the action(s) to perform. That is, each facet argues for or against motivations. The facet objectives consist of promoting (or demoting) the goals to reach, the actions to performed (or not) and the beliefs to grant (or not) in order to actively achieve some aspects of the agent's agenda or avoid some situations that would be detrimental. In addition, facets monitor the statements proposed by other ones in $\mathtt{AS}$ to determine whether these statements interact with their own agenda. Essentially each facet can have an opinion of what is best for the agent as a whole, but from its limited viewpoint. A facet must argue its case against/with other possibly competing/completing views for this to become incorporated into the agent overt behavior. This behavior is determined by the *personality* that specifies how to give priority to the facets and arbitrate amongst them to resolve the possible conflicts.

4 ARGUMENTATION FRAMEWORK

Our argumentation framework (AF) is based on the opposition calculus of (Dung, 1995), where arguments are reasons which can be defeated by other arguments.

Definition 1 (AF). *An* argumentation framework *is a pair* $AF = \langle \mathcal{A}, \texttt{defeats} \rangle$ *where* $\mathcal{A}$ *is a finite set of arguments and* $\texttt{defeats}$ *is a binary relation over* $\mathcal{A}$*. We say that an argument* b *defeats an argument* a *if* $(b,a) \in \texttt{defeats}$*. Additionally, we say that a set of arguments* S *defeats* a *if* $(b,a) \in \texttt{defeats}$ *and* $b \in S$*.*

(Dung, 1995) analysis if a set of arguments is collectively justified.

Definition 2 (Semantics). *A set of arguments* $S \subseteq \mathcal{A}$ *is:*

- conflict-free *if* $\forall a, b \in S$ *it is not the case that* a *defeats* b*;*
- admissible *if* S *is conflict-free and* S *defeats every argument* a *which defeats some arguments in* S*.*

Amongst the semantics proposed in (Dung, 1995), we restrict ourself to the admissible one.
We use AF to model the reasoning within the V3A agent architecture.

Definition 3 (KF). *A* knowledge representation framework *is a tuple* $KF = \langle \mathcal{L}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, P \rangle$ *where:*

- $\mathcal{L}$ *is a formal language consisting of a finite set of sentences, called the* representation language*;*
- $\mathcal{A}sm$ *is a set of atoms in* $\mathcal{L}$*, called* assumptions*, which are taken for granted;*
- $\mathcal{I}$ *is a binary relation over atoms in* $\mathcal{L}$*, called the* incompatibility relation*, which is asymmetric;*
- $\mathcal{T}$ *is a finite set of rules built upon* $\mathcal{L}$*, called the* theory*;*
- $P \subseteq \mathcal{T} \times \mathcal{T}$ *is a transitive, irreflexive and asymmetric relation over* $\mathcal{T}$*, called the* priority relation*.*

$\mathcal{L}$ admits strong negation (classical negation) and weak negation (negation as failure). A strong literal is an atomic first-order formula, possible preceded by strong negation $\neg$. A weak literal is a literal of the form $\sim L$, where L is strong.
We adopt an assumption-based argumentation approach (Dung et al., 2007) to reason about beliefs, goals, decisions and priorities (Morge and Mancarella, 2007). That is, agents can reason under uncertainty. Actually, certain literals are *assumable*, meaning that they can be assumed to hold in the KB as long as there is no evidence to the contrary. Decisions (e.g. $\texttt{request}(\texttt{psa}, \texttt{sa}, ticket)$) as well as some beliefs (e.g. $\sim \texttt{strike}$) are assumable literals.
The *incompatibility relation* captures conflicts. We have $L \ \mathcal{I} \ \neg L$, $\neg L \ \mathcal{I} \ L$ and $L \ \mathcal{I} \sim L$. It is not the case that $\sim L \ \mathcal{I} \ L$. For instance, $\texttt{paycom}(\texttt{psa}, \texttt{sa}, price) \ \mathcal{I} \ \neg\texttt{paycom}(\texttt{psa}, \texttt{sa}, price)$ and $\neg\texttt{paycom}(\texttt{psa}, \texttt{sa}, price) \ \mathcal{I} \ \texttt{paycom}(\texttt{psa}, \texttt{sa}, price)$ whatever the price is. We say that two sets of sentences Φ_1 and Φ_2 are incompatible ($\Phi_1 \ \mathcal{I} \ \Phi_2$) iff there is at least one sentence ϕ_1 if Φ_1 and one sentence ϕ_2 in Φ_2 such that $\phi_1 \ \mathcal{I} \ \phi_2$.

A *theory* is a collection of rules with priorities over them.

Definition 4 (Theory). *A* theory $\mathcal{T}$ *is an extended logic program, i.e a finite set of rules s.t.* R*:* $L_0 \leftarrow L_1, \ldots, L_j, \sim L_{j+1}, \ldots, \sim L_n$ *with* $n \geq 0$*, each* L_i *being a strong literal in* $\mathcal{L}$*. The literal* L_0*, called* head *of the rule (denoted* $\texttt{head}(R)$*), is a statement. The finite set* $\{L_1, \ldots, \sim L_n\}$*, called* body *of the rule, is denoted* $\texttt{body}(R)$*.* R*, called* name *of the rule, is an atom in* $\mathcal{L}$*. All variables occurring in a rule are implicitly universally quantified over the whole rule. A rule with variables is a scheme standing for all its ground instances.*

For simplicity, we will assume that the names of rules are neither in the bodies nor in the head of the rules thus avoiding self-reference problems. We consider the *priority* relation P on the rules in $\mathcal{T}$, which is transitive, irreflexive and asymmetric. $R_1 P R_2$ can be read "R_1 has priority over R_2". There is no priority between R_1 and R_2, either because R_1 and R_2 are *ex æquo*, or because R_1 and R_2 are not comparable. For instance, $\texttt{user}(f_1, x_1) P \texttt{agt}(f_2, x_2))$ means that the rules in $\texttt{KB}_{\texttt{user}}$ have priority over the rules in $\texttt{KB}_{\texttt{agt}}$.

The KBases and the personality of the PSA in Pisa are depicted in Tab. 1.
Users KB. The user desires to reach the next travel step without paying any extra commissions.
Agents KB. The PSA knows that the SA can deliver tickets after requesting him. Contrary to Italy, utilising the services of a SA in UK requires to pay an extra commission. The cost of tickets also depends on the context.
Environments KB. These facets work on percepts

e.g. the time and the location of the PSA (cf $\mathtt{env}(\mathtt{f}_4, \mathtt{a}_2)$) and beliefs about the instant location, e.g. the time schedule (cf $\mathtt{env}(\mathtt{f}_3, \mathtt{r}_1)$).

Interactions KB. These facets also work on percepts e.g. messages which will be received/sent and beliefs about protocols which depends on the context e.g. $\mathtt{int}(\mathtt{f}_6, \mathtt{r}_1(aid_1, aid_2, ticket, price, comm))$.

Organisations KB. The PSA must pay a ticket to confirm its reservation.

Personality. Since the PSA has been set up by the constructor to respect the user's preferences and assist him, the facets related to the user are preferred than those that are related to other agents. In order to utilise the computing system, the PSA prefers the organisation facet rather the ones which are related to the environment. Contrary to other components, the personality is not embodied by facets but these preferences are encoded in the procedural rules described informally in Section 5.

In this scenario, self-adaptation is crucial since, when Max is arriving in a new location, new KBases can be downloaded and they replace the previous one. It is worth noticing that the users KB and the personality are not modified.

Due to the abductive nature of proactive agent reasoning, arguments are built by reasoning backward.

Definition 5 (Argument). *An* argument a *for a statement* $\alpha \in \mathcal{L}$ *(denoted* $\mathtt{conc}(a)$*) is a deduction of that conclusion whose premise is a set of rules (denoted* $\mathtt{rules}(a)$*) and assumptions (denoted* $\mathtt{asm}(a)$*) of* KF.
The top-level link *of* a *(denoted* $\top(a)$*) is a rule s.t. its head is* $\mathtt{conc}(a)$.
The sentences *of* a *(denoted* $\mathtt{sent}(a)$*) is the set of literals of* $\mathcal{L}$ *in the bodies/heads of the rules including the assumptions of* a.

In Pisa, the argument a concludes motive since the PSA does not register. This argument is defined.t.
$\mathtt{rules}(\mathtt{a})=\{\mathtt{user}(\mathtt{f}_1,\mathtt{r}_1),\mathtt{env}(\mathtt{f}_3,\mathtt{r}_1),$
$\mathtt{env}(\mathtt{f}_4,\mathtt{a}_1),\mathtt{env}(\mathtt{f}_4,\mathtt{a}_2),\mathtt{env}(\mathtt{f}_3,\mathtt{r}_3(\mathtt{lipifi}))\}$,
$\top(\mathtt{a}) = \mathtt{user}(\mathtt{f}_1,\mathtt{r}_1)$, and
$\mathtt{asm}(\mathtt{a}) = \{\sim \mathtt{strike}, \mathtt{register}(\mathtt{lipifi},\mathtt{no}),$
$\mathtt{be}(\mathtt{pisa},17)\}$.
By contrast, the argument b concludes motive if the PSA request a ticket, pays it and registers. This argument is defined s.t.
$\mathtt{rules}(\mathtt{b})=\{\mathtt{user}(\mathtt{f}_1,\mathtt{r}_1),\mathtt{env}(\mathtt{f}_3,\mathtt{r}_1),$
$\mathtt{env}(\mathtt{f}_3,\mathtt{r}_2(\mathtt{lipifi})),\mathtt{org}(\mathtt{f}_7,\mathtt{r}_1(\mathtt{lipifi},ticket,price,comm)),$
$\mathtt{agt}(\mathtt{f}_2,\mathtt{r}_1(ticket)),\mathtt{int}(\mathtt{f}_6,\mathtt{r}_1(\mathtt{sa},\mathtt{psa},ticket,price,comm))$
$\}$
$\top(\mathtt{b}) = \mathtt{user}(\mathtt{f}_1,\mathtt{r}_1)$, and
$\mathtt{asm}(\mathtt{b}) = \{\sim \mathtt{strike}, \mathtt{be}(\mathtt{pisa},17), \mathtt{pay}(\mathtt{psa},\mathtt{sa},4,0),$
$\mathtt{request}(\mathtt{psa},\mathtt{sa},ticket)\}$. After self-adaptation, similar arguments exist in Stansted.

We define here the defeat relation. Firstly, we define the attack relation to deal with conflicting arguments.

Definition 6 (Attacks). *Let* a *and* b *be two arguments.* a attacks b *iff* $\mathtt{sent}(a) \; \mathcal{I} \; \mathtt{sent}(b)$.

This relation encompasses both the *rebuttal* attack due to the incompatibility of conclusions, and the *undermining* attack, i.e. directed to a "subconclusion". The strength of arguments depends on the priority of their sentences. In order to give a criterion that will allow to prefer one argument over another, we consider here the last link principle to promote high-level goals.

Definition 7 (Strength). *Let* a *and* b *be two arguments.* a *is* stronger *than* b *(denoted* $\mathtt{prior}(a,b)$*) if it is the case that* $\top(a) P \top(b)$.

The two previous relations can be combined.

Definition 8 (Defeats). *Let* a *and* b *be two arguments.* a defeats b *iff: i)* a $\mathtt{attacks}$ b*; ii) it is not the case that* $\mathtt{prior}(b,a)$.

In Pisa, $\{\mathtt{b}\}$ is in an admissible set since the organisation has priority over the environment. This argument describes the motivation for registering. After the self-adaptation of the PSA in London, even if a new travel connection is considered, this argument is no more admissible since an extra commission is required. Due to the agent personality, argument a is reinstantiated and the PSA does not register.

5 DIALOGUE-GAME

The result of the debate amongst facets is an argument sketched in AS. We consider here the procedural rules which regulate the exchanges of moves to reach an agreement. For this purpose, we instantiate a dialectical framework (Prakken, 2006).

Definition 9 (Dialectical Framework). *Let us consider the* $\mathtt{topic}$*, i.e. a statement in* $\mathcal{L}$*, and* $\mathcal{FCL}$ *a facet communication language. The* dialectical framework *is a tuple* $DF(\mathtt{topic}, KF) = \langle P, AS, \Omega_M, H, T, \mathtt{proto}, Z \rangle$ *where:*

- $P = \{p_1, \ldots, p_n\}$ *is a set of n players;*
- $AS = \langle \mathtt{Pro}, \mathtt{Opp} \rangle$ *is composed of two* boards $\mathtt{Pro}$ *and* $\mathtt{Opp}$ *which contains the literals held by the proponents and the opponents respectively;*
- $\Omega_M \subseteq \mathcal{FCL}$ *is a set of well-formed moves;*
- H *is a set of histories, the sequences of well-formed moves s.t. the speaker of a move is determined at each stage by the turn-taking function* T *and the moves agree with the protocol* $\mathtt{proto}$*;*

Table 1: The KBases of the PSA in Pisa.

	$\mathcal{T}$	$\mathcal{A}sm$
KB_{user}	$user(f_1,r_1)$: $motive \leftarrow be(apisa,18)$ $user(f_1,r_2)$: $motive \leftarrow be(london,22)$	$\neg pay(psa,aid,price,comm)$ with $comm \neq 0$
KB_{agt}	$agt(f_2,r_1(ticket))$: $buy(psa,sa,ticket,4,0) \leftarrow accept(psa,sa,ticket,4,0)$	
KB_{env}	$env(f_3,r_1)$: $be(apisa,18) \leftarrow be(pisa,17), \sim strike, take\_train(lipifi)$ $env(f_3,r_2(train))$: $take\_train(train) \leftarrow register(train,yes)$ $env(f_3,r_3(train))$: $take\_train(train) \leftarrow register(train,no)$	$register(train,no)$
		$\sim strike$ $be(pisa,17)$
KB_{int}		$request(psa,aid_1,ticket)$
	$int(f_6,r_1(aid_1,aid_2,ticket,price,comm))$: $accept(aid_1,aid_2,ticket,price,comm) \leftarrow request(aid_2,aid_1,ticket)$ $int(f_6,r_2(aid_1,aid_2,ticket,price,comm))$: $\neg accept(aid_1,aid_2,ticket) \leftarrow request(aid_2,aid_1,ticket)$	
KB_{org}	$org(f_7,r_1(train,ticket,price,comm))$: $register(train,yes) \leftarrow buy(psa,sa,ticket,price,comm), pay(psa,sa,price,comm)$ $org(f_7,r_2(train,ticket,price,comm))$: $\neg register(train,yes) \leftarrow \neg buy(psa,sa,ticket,price,comm)$ $org(f_7,r_3(train,ticket,price,comm))$: $\neg register(train,yes) \leftarrow buy(psa,sa,ticket,price,comm), \neg pay(psa,sa,price,comm)$	$pay(psa,sa,price,comm)$
pers	$user(f_1,x_1) P agt(f_2,x_2)$ $org(f_1,x_1) P env(f_2,x_2)$	$int(f,x)$, $org(f,x)$, $agt(f,x)$, $env(f,x)$, $user(f,x)$

- *T: $H \rightarrow P$ is the turn-taking function;*
- *$proto$: $H \times AS \rightarrow \Omega_M$ is the function determining the moves which are allowed to expand an history;*
- *Z is the set of dialogues, i.e. the terminal histories where the proponent (respectively opponent) board is a set of assumable literals (respectively empty).*

In the V3A architecture, the DF allows multi-party dialogues amongst facets (the players) about motive (the topic) within KF. Players claim literals during dialogues. Each dialogue is a maximally long sequence of moves. We call *line* the sub-sequence of moves where backtracking is ignored. Amongst players, the proponents argue for an initial claim while the opponents argue against it.

We define here the syntax and the semantics of moves. The *syntax* of moves is in conformance with a common *facet communication language*, $\mathcal{FCL}$. A move at time t: has an identifier, mv_t; is uttered by a speaker ($sp_t \in P$); eventually rp_t is the identifier of the message to which mv_t responds and the speech act is composed of a locution loc_t and a content $content_t$. The locutions are claim, concede, oppose, deny, and unknown. The content is a set of atoms in $\mathcal{L}$.

The *semantics* of speech acts is public since all players confer the same meaning to the moves. The semantics is defined in terms of pre/post-conditions.

Definition 10 (**Semantics of $\mathcal{FCL}$).** *Let t be the time of a history h in H ($0 \leq t < |h|$). $AS_0 = \langle\{topic\}, \emptyset\rangle$. The* semantics *of the utterance by the facet f at time t is defined s.t.:*

1. *f may utter $unknown(\emptyset)$ and so $AS_{t+1} = AS_t$;*
2. *considering $L \in Pro_t$,*
   (a) *f may utter $claim(P)$, if $\exists r \in \mathcal{T}_f$ $head(r) = L$, $body(r) = P$, $P \cap Pro_t = \emptyset$, and it is not the case that $P \; \mathcal{I} \; Pro_t$. Therefore, $AS_{t+1} = \langle Pro_t \cup P - \{L\}, Opp_t\rangle$,*
   (b) *f may utter $concede(\{L\})$, if $L \in \mathcal{A}sm_f$. Therefore, $AS_{t+1} = AS_t$,*
   (c) *f may utter $oppose(\{L'\})$, if $L' \; \mathcal{I} \; L$. Therefore, $AS_{t+1} = \langle Pro_t - \{L\}, Opp_t \cup \{L'\}\rangle$;*
3. *considering $L \in Opp_t$,*
   (a) *f may utter $claim(P)$, if $\exists r \in \mathcal{T}_f$ $head(r) = L$, $body(r) = P$ and $P \cap Opp_t = \emptyset$. Therefore, $AS_{t+1} = \langle Pro_t, Opp_t \cup P - \{L\}\rangle$,*
   (b) *f may utter $deny(\{L'\})$, if $L' \; \mathcal{I} \; L$. Therefore, $AS_{t+1} = \langle Pro_t \cup \{L'\}, Opp_t - \{L\}\rangle$.*

The rules to update AS incorporates a filtering (in case 2a and 3a) to be more efficient. Concretely, the set of literals in Pro and Opp are filtered, so they are not repeated more than once, and finally the literals in Pro are not incompatible with each other. The speech act unknown($\emptyset$) has no preconditions. Neither concessions nor pleas of ignorance have effect on AS.

In order to be uttered, a move must be *well-formed*. The initial moves are initial claims and pleas of ignorance: $mv_0 \in \Omega_M$ iff $loc(mv_0) = claim$ or $loc(mv_0) = unknown$. The replying moves are well-formed iff they refer to an earlier move: $mv_j \in \Omega_M$ iff $rp_j = mv_i$ with $0 \leq i < j$. Notice that backtracking is allowed. Each dialogue is a sequence $h = (mv_0, \ldots, mv_{|h|-1})$ with $proto(h) = \emptyset$. In this way, the set Z of dialogues is a set of maximally long histories, i.e. which cannot be expanded even if backtracking is allowed.

The *turn-taking function* T determines the speaker of each move. If $h \in H$, $sp_0 = p_i$ and $j - i = |h|(mod\ n)$, then $T(h) = p_j$.

The protocol (proto) consists of a set of sequence rules (e.g. $sr_1, \ldots sr_4$ represented in Tab. 2) specifying the legal replying moves. For example, sr_1 specifies the legal moves replying to a previous claim ($claim(P)$). The speech acts resist or surrender, i.e. close the line. Players resist as much as possible. The locutions concede and unknown are utilised to manage the sequence of moves since they surrender, and so close the line but not necessarily the dialogue (backtracking is allowed). By contrast, a claim ($claim(P')$) and an opposition ($oppose(\{L'\})$) resist to the previous claim. The moves replying to a deny ($deny(\{L'\})$) are the same as the replying move of a

Table 2: Speech acts and the potential replies.

	Speech acts	Resisting	Surrendering
$\mathtt{sr}_1$	$\mathtt{claim}(P)$	$\mathtt{claim}(P')$, with $\mathtt{r}$ s.t $L = \mathtt{head}(\mathtt{r}) \in P$ and $\mathtt{body}(\mathtt{r}) = P'$	$\mathtt{concede}(\{L\})$, with $L \in P$
		$\mathtt{oppose}(\{L'\})$, with $L' \mathit{I} L$ and $L \in P$	$\mathtt{unknown}(P)$
$\mathtt{sr}_2$	$\mathtt{oppose}(\{L\})$	$\mathtt{claim}(P')$, with $\mathtt{r}$ s.t. $L = \mathtt{head}(\mathtt{r})$ and $\mathtt{body}(\mathtt{r}) = P'$	$\mathtt{unknown}(P)$
		$\mathtt{deny}(\{L'\})$, with $L' \mathit{I} L$	
$\mathtt{sr}_3$	$\mathtt{concede}(P)$	$\emptyset$	$\emptyset$
$\mathtt{sr}_4$	$\mathtt{unknown}(P)$	$\emptyset$	$\emptyset$

claim ($\mathtt{claim}(\{L'\})$). At the end of the game, `Pro` may contains the assumptions of an argument deducing `motive`.

6 CONCLUSIONS

We have proposed a dialectical argumentation framework allowing an agent to argue with itself about its motivations. The framework relies upon the admissibility semantics and uses an assumption-based argumentation approach to support reasoning about the knowledge, goals, and decisions held in the agent's mental facets. These modules interact via a dialogue-game which is formally defined and exemplified via a concrete scenario. The contribution of the work is a modular model that allows the facets and the personality of an agent to be specified declaratively, manages potential conflicts and replaces components at runtime, thus avoiding to restart the agent's reasoning process whenever a component joins or leaves the game.
Some of the concepts utilized here have been introduced in the AAA model (Witkowski and Stathis, 2004). However, here we provide a formal definition of the argumentation game that the original AAA model abstracted away from. We have also reinterpreted the original model by using the Vowels approach, which has an agent-oriented software engineering foundation.
Our work is also related to the KGP (Kakas et al., 2004b) model of agency and in particular the modelling of the personality of the agent (Kakas et al., 2004a) through preferences. One important difference with (Kakas et al., 2004a) comes from our decomposition which distinguishes explicitly the different aspects, possibly conflicting, that the agent must arbitrate. These aspects are embodied by faculties that are more amenable to be plug-and-play components at run-time using a multi-threaded implementation.

Future work includes investigating the properties of different dialogue-games for different semantics and properties. We also plan to extend the current prototype using CaSAPI [1] to allow an internal dialectic that is multi-threaded and relies on facets that are interpreted by different proof systems implementing different kinds of reasoning such as epistemic reasoning, practical reasoning and normative reasoning.

ACKNOWLEDGEMENTS

This work is supported by the Sixth Framework IST 035200 ARGUGRID project.

REFERENCES

Demazeau, Y. (1995). From interactions to collective behaviour in agent-based systems. In *Proc. of the First European Conference on Cognitive Science*, pages 117–132, Saint Malo.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357.

Dung, P. M., Mancarella, P., and Toni, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence, Special Issue on Argumentation*, 171(10-15):642–674.

Kakas, A. C., Mancarella, P., Sadri, F., Stathis, K., and Toni, F. (2004a). Declarative agent control. In Leite, J. A. and Torroni, P., editors, *CLIMA V*, volume 3487 of *LNCS*, pages 96–110.

Kakas, A. C., Mancarella, P., Sadri, F., Stathis, K., and Toni, F. (2004b). The KGP model of agency. In *Proc. of ECAI*, pages 33–37.

Meurisse, T. and Briot, J.-P. (2001). Une approche à base de composants pour la conception d'agents. *Technique et Science Informatiques (TSI)*, 20(4).

Morge, M. and Mancarella, P. (2007). The hedgehog and the fox. An argumentation-based decision support system. In *Proc. of ArgMAS*, pages 55–68.

Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21:163–188.

Ricordel, P.-M. and Demazeau, Y. (2000). From analysis to deployment: A multi-agent platform survey. In *Proc. of ESAW*, volume 1972/2000 of *LNCS*, pages 93–105, Berlin, Germay. Springer Berlin / Heidelberg.

Vercouter, L. (2004). MAST: Un modèle de composants pour la conception de SMA. In *Actes de JMAC'04*, Paris, France.

Witkowski, M. and Stathis, K. (2004). A dialectic architecture for computational autonomy. In *Agents and Computational Autonomy*, pages 261–273. Springer Berlin.

[1] http://www.doc.ic.ac.uk/~dg00/casapi.html