Journal of Universal Computer Science, vol. 15, no. 13 (2009), 2676-2700
submitted: 31/10/08, accepted: 13/6/09, appeared: 1/7/09 © J.UCS

Analyzing Cooperation in Iterative Social Network Design

Guido Boella
(University of Turin, Italy
guido@di.unito.it)

Leendert van der Torre
(University of Luxembourg, Luxembourg
leendert@vandertorre.com)

Serena Villata
(University of Turin, Italy
villata@di.unito.it)

Abstract: We introduce an approach to iteratively design 'small' social networks used in software engineering together with methods analyzing the cooperation in the system. The degree of cooperation is measured by the emergence of coalitions and their stability over time. At the most abstract level, which we call the coalition view, coalitions are abstract entities that may dominate or attack other coalitions. During iterative design, these abstract entities are refined with agents and their dependencies constituting the coalitions (dependence view), the powers of sets of agents to see to goals (power view) and finally the beliefs, plans, tasks and goals of agents (agent view). The analysis methods predict the emergence of coalitions based on reciprocity and argumentation theory.

Key Words: Coalitions, multiagent systems, argumentation, dependence networks.
Category: I.2.11, I.6.5

1 Introduction

A social network is a social structure composed by nodes, which are generally individuals or organizations, that are tied by one or more specific types of interdependency. Wide social networks and small ones share the same structure but different kinds of analysis are needed. The analysis of large social networks [Hanneman and Riddle 2005] is usually based on either data-mining or graph-based techniques, such as small world properties, centrality, cliques, similarity, and so on. These analysis tools work well for large networks, such as those composed by the nodes in the world wide web or the members of FaceBook, but they work less well for small networks representing the relations among stakeholders in software engineering. Moreover, they do not support iterative design of software in order to interact with the designed system to provide a form of research for informing and evolving a project, as successive versions.

Small social networks are analyzed in software engineering, for example by the TROPOS methodology [Bresciani et el. 2004], developed for agent-oriented design of software systems. The intuition of the TROPOS methodology [Bresciani et el. 2004] is to couple the instruments offered by software engineering and the multiagent paradigm. In this paradigm, the entities composing the system are agent, autonomous by definition, characterized by their own sets of goals, capabilities and beliefs. The multiagent paradigm allows the cooperation among the agents with the aim to obtain common and personal goals. In this way, multiagent systems offer a solution for open, distributed and complex systems and the approach combining software engineering and multiagent systems is defined Agent-Oriented Software Engineering. A typical social dependence network in the TROPOS methodology [Bresciani et el. 2004] contains at most a hundred nodes, in contrast to the hundreds of thousands of nodes used in the web or in FaceBook.

In this paper, we are interested in the analysis based on cooperation which emerges in 'small' social networks in order to achieve a greater number of goals. As a measure of cooperation, we analyze the coalitions [Shehory and Kraus 1998] that emerge in a social network assuming reciprocity, for example measuring the number of coalitions [van der Torre and Villata 2009], the kinds of coalitions [Boella et al. 2008], or the stability of the coalitions. This breaks down in the following questions.

1. How to iteratively design a social network?

2. How to analyze the reciprocity based coalitions that may emerge in social networks at various degrees of abstraction?

3. How to refine the abstract coalition models with social dependencies among agents, powers of sets of agents, and plans or tasks?

At the highest level of abstraction, coalitions are purely abstract and we only specify whether the creation of one coalition will block the creation of another coalition. We say that two coalitions are attacking each other and the stability argument sets a preference of the first coalition over the second one, and we use abstract argumentation theory [Bench-Capon and Dunne 2007] to determine the acceptable coalitions. At the second level of abstraction, we detail the composition of a coalition which is seen as a set of agents and a set of dependencies between them. Our notion of coalition is based on the concept of reciprocity which constraints each node to contribute something, and to get something out of it. For example, in a virtual organization each node has to be useful for another node. At the third level of abstraction, we detail the powers and goals of the individual agents. At the fourth level of abstraction, we also detail the beliefs, decisions and goals of the agents. For the analysis we focus on the coalition and dependence views, and leave a detailed analysis of the power and agent views for further research.

We illustrate our approach using a grid scenario. Consider, for example, a virtual organization for e-Science composed by nodes belonging to academic institutions such as universities and research centers. Inside the virtual organization, sub-groups can be formed with the aim to collaborate in order to achieve a greater number of goals, i.e., if node a cannot store a file but it can help node b in doing a computation and b can store a's file, these two nodes form a reciprocity based coalition in order to achieve both goals. It would be possible that two or more candidate coalitions share the same goals, e.g. two nodes can do the storage for node a and thus it becomes necessary to have a mechanism to decide what coalition can be formed.

Using social dependence networks to represent the multiagent system, as in TROPOS [Bresciani et el. 2004], allows us to model, particularly for the requirements analysis phase of the design process, the domain stakeholders. The analysis of cooperation in this context is relevant since agents can form coalitions with the aim to achieve more goals than what they can achieve alone. As in well known game theoretic approaches to cooperation [Shehory and Kraus 1998], we face with problems of incompatibilities between the possible coalitions which can be formed. We manage these incompatibilities using an argumentation framework treating each candidate coalition as an argument, the incompatibilities as the attacks between the arguments and, finally, using the extensions to find out the acceptable coalitions.

The layout of this paper is as follows. Section 2 presents a grid-based scenario showing how the presented methods can be used to model real systems. In Section 3, we present how to argue about the coalition view level using an argumentation framework. Section 4 presents how to argue about the dynamic dependence view level to know which coalitions are formed. In Section 5, we present the agent view and the power view. Related work and conclusions end the paper.

2 Iterative social network design

In this section, we answer to the first research question, presenting the four viewpoints composing our iterative design model and we describe the concepts we use in the model thanks to an ontology. Moreover, we provide a running scenario based on the grid architecture explaining our model of iterative design for small social networks.

2.1 Coalitions in a grid-based scenario

Cooperation in grid, in particular virtual organizations, can be seen as coalition formation in social networks. A virtual organization allows the users, their roles and the resources they can access in a collaborative project to be defined

[Foster et al. 2001]. In particular, we look into small sets of nodes within virtual organizations as coalitions. Reciprocity-based coalitions can be viewed as subsets of a virtual organization, in which there is the constraint that each node has to contribute something, and has to get something out of it.

The scenario of virtual organizations based on grid networks represents a case study able to underline the benefits of the presented viewpoints and the argumentation framework to argue about the evolution of coalitions over time. First of all, in the multiagent paradigm agents' autonomy is assumed in all representations, i.e., the grid philosophy imposes the autonomy of the nodes composing it. Second, the presented model depicts the system using dependence networks, structures similar to the grid network itself. Finally, the idea that subsets of nodes composing a virtual organization compose also different coalitions sharing common goals and attacking each others helps in providing the intuition of the addressed problem and the proposed solutions.

Concerning viewpoints, a virtual organization can be represented using our four views in order to highlight different aspects: the agent view presents each node of the grid as an agent with a set of associated skills and goals, the power view presents the nodes which can achieve the goals of the virtual organization and what are the nodes with the conditional power to add new goals to the other nodes, the dynamic dependence view describes the virtual organization in terms of dependencies giving it a network structure and, finally, the coalition view represents the virtual organization as sets of nodes representing reciprocity-based groups. In this context, the modeled stakeholders represent the nodes of the virtual organization and their concern is to store and run data.

2.2 Ontology

In this section, we introduce the ontology used in our model, represented as a UML diagram shown in Figure 1. This ontology summarizes the concepts introduced in our four views. Particularly, it introduces the concepts of agent, fact, skill and goal. Each agent has a set of facts in which it believes and a set of goals that it has to achieve by means of its skills and these relations are represented by the agent view. Figure 1 presents two kinds of dependencies, dependencies and dynamic dependencies. The first one explain that an agent (depender) depends on another agent (dependee) to achieve a goal (dependum) while dynamic dependencies enable the addition or removal of dependencies by a third agent (dyndep creator). The notion of coalition, with its subclasses, is linked to both the concepts of common and dynamic dependency and agent since we define a coalition as a set of dependencies and agents. The preference of one coalition over the other one is represented by the higher order dependency which is a dynamic dependency. Finally, we introduce in our four views the concept of time representing the agents, the dependencies and the coalitions present in the system in each time instant.

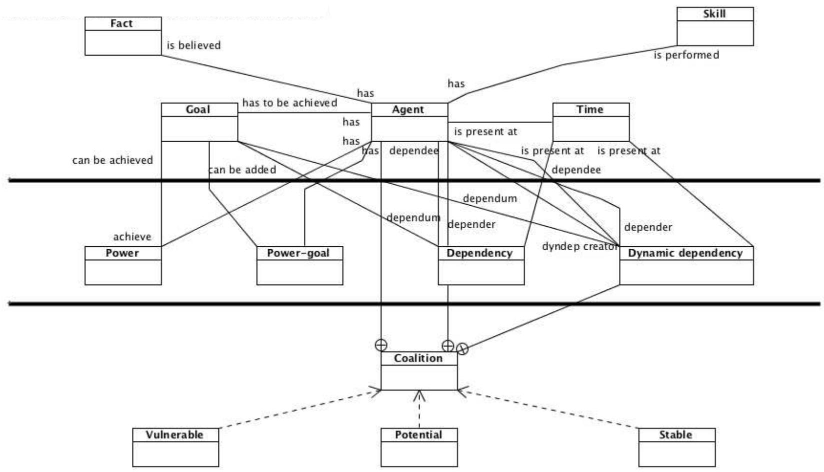

Figure 1: UML diagram of the ontology of our model.

2.3 Iterative design: refining viewpoints on grid-based coalitions

Figure 2 illustrates the iterative design of the grid scenario. It contains our four viewpoints and the refinement relations between them. Each row in the table explains one viewpoint. Going from one row to the one below is a refinement, and going to a row above is an abstraction. The designer starts with the top row, and refines it in each step to the row below it. It can well be that the designer encounters a problem in a more refined view and then has to adapt the more abstract views, leading to the iterative design cycle. However, here we consider only the refinements of the views, not the revisions or updates of them.

n this section, we describe the four viewpoints in detail. For each viewpoint represented by a row, the leftmost column summarizes the part of the ontology used for this viewpoint. The next two columns visualize the first two elements of the temporal sequence within the viewpoint. The rightmost column gives some additional explanation on the grid example. The analysis method is implicitly represented in the example. Cooperation is represented by straight and dashed lines. A straight line represents the candidate coalition, and a dashed line represents that it is not formed.

The coalition view, in Figure 3, represents the most abstract viewpoint used to argue on coalitions. Concepts used in this viewpoint are two kinds of nodes,

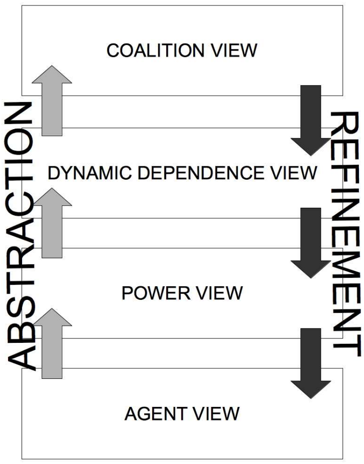

Figure 2: Iterative social network design.

called coalitions and stability argument, and one kind of relation, called dominance or attack. The attack relation between candidate coalitions influences which coalition will be formed. In the grid example, we distinguish two candidate coalitions, formed by nodes of a virtual organization, attacking each other, and one stability argument, preferring the first candidate coalition over the second one. This stability argument attacks the attack from the candidate coalition C_2 to the candidate coalition C_1 at time instant t_1, and this second-order attack leads to the formation of coalition C_1. The stability argument can itself be attacked by another stability argument in an higher order attack, not represented in the figure.

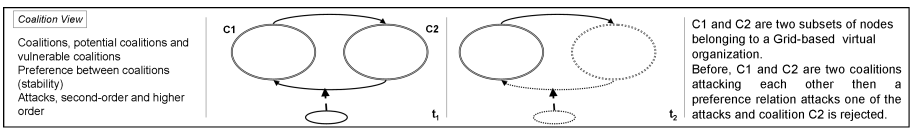

Figure 3: Coalition view.

The dynamic dependence view, in Figure 4, represents a refinement of the coalition view, because we introduce the agents and the dependencies that constitute the coalitions. Concepts used in this viewpoint are one kind of node, agents, and two kinds of relations, representing respectively simple dependen-

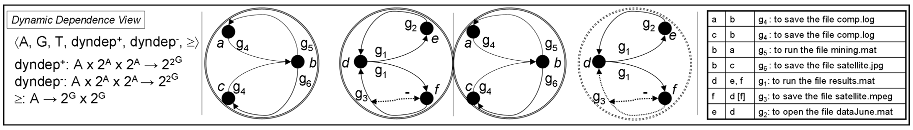

Figure 4: Dynamic dependence view.

cies and higher order dependencies. The goals are represented only as labels of the dependence relations. In the grid example, each coalition consists of three nodes. A node can depend on two other nodes for the same goal, as in the case of node d for goal g_1 or two nodes can depend on the same node for a shared goal, as in the case of nodes a and c for goal g_4. The dynamic dependency of the example sees node f able to delete the dependency between itself and node d concerning goal g_3.

In the power view, in Figure 5,we refine the dynamic dependence view. Concepts used in this viewpoint are the same nodes as before, agents and goals, but three new relations, one associating agents with goals (goals), one which says which goals a *set of* agents can achieve (power), and one which represents which sets of goals can be created or destroyed by an agent (power-goal). Likewise there is the possibility to create or destroy powers, not directly represented in the figure. The power relation is depicted as a square including agents and goals and the power-goal relation is depicted as a squared goal linked to the agents that can add or remove it. In the grid example, node f has the power-goal to delete its goal g_3 while node d has the power to see to g_3.

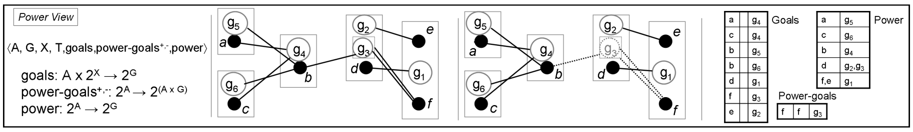

Figure 5: Power view.

In the agent view, in Figure 6, we finally refine the power view. The used concepts are skills and rules. Each agent has some skills, whereas in the power view, each *set* of agents has power. So the power view is more "social" than the agent view. In Figure 2, skills are represented for each agent whereas the power is represented for a set of agents, as indicated by the square around them. The agent view is the most detailed view since it considers all the features of the single agents but it looses the notion of "group" in the power view.

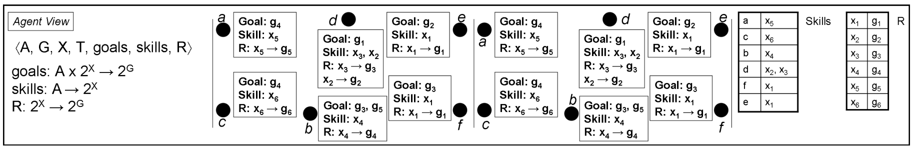

Figure 6: Agent view.

3 Arguing on abstract coalitions models

In this section, we answer to the second research question presenting the abstract coalitions models on which we analyze reciprocity based coalitions that may emerge in social networks at the higher level of abstraction. This can be specified by the following subquestion: How to represent coalition formation and coalitional game theory in Dung's argumentation theory? [Dung 1995] introduces game theory as one of the three applications of his abstract theory (besides non-monotonic reasoning and logic programming), and [Amgoud 2005] shows how to instantiate preference-based argumentation with a task-based coalition formation theory. However, in [Amgoud 2005], arguments why one coalition would be preferred over another one are not open for debate. In our approach, the preference between arguments is defined in terms of the stability argument. This additional argument sets the preference of one arguments over the others, attacking the attacks towards the preferred arguments. The name stability argument is used to express coalitions' evolution where, on the one hand, coalition's stability is maintained if the coalition is not attacked by the other coalitions, and, on the other hand, the stability is destroyed if the coalition is not preferred over the others and thus it is attacked by some other coalitions.

3.1 Dung's abstract argumentation framework

We follow [Baroni and Giacomin 2007]. An underlying mechanism of argument generation defines a set of arguments, which is typically infinite, and which we call the universe of arguments and represent by $\mathcal{U}$. An acceptance function $\mathcal{E}$ is a function that associates with a set of arguments $\mathcal{B} \subseteq \mathcal{U}$, a set of arguments produced by a reasoner at a given time instant, and a binary relation $\rightarrow \subseteq \mathcal{B} \times \mathcal{B}$, representing the dominance or attack relation among these arguments. We use this acceptance function to obtain the acceptable arguments due to the chosen acceptability semantics.

Definition 1. Let $\mathcal{U}$ be the universe of arguments. An acceptance function $\mathcal{E} : 2^{\mathcal{U}} \times 2^{\mathcal{U}\times\mathcal{U}} \rightarrow 2^{2^{\mathcal{U}}}$ is a partial function which is defined for each argumentation framework $\langle \mathcal{B}, \rightarrow \rangle$ with finite $\mathcal{B} \subseteq \mathcal{U}$ and $\rightarrow \subseteq \mathcal{B} \times \mathcal{B}$, and which associates with argumentation framework $\langle \mathcal{B}, \rightarrow \rangle$ sets of subsets of $\mathcal{B}$: $\mathcal{E}(\langle \mathcal{B}, \rightarrow \rangle) \subseteq 2^{\mathcal{B}}$.

Baroni and Giacomin identify the following two fundamental principles underlying the definition of extension-based semantics in Dung's framework, the *language independent* principle and the *conflict free* principle. The notion of conflict free is provided below. A further discussion on these principles is provided in [Baroni and Giacomin 2007]

Definition 2 Conflict free. Given an argumentation framework $\mathcal{AF} = \langle \mathcal{B}, \rightarrow \rangle$, a set $S \subseteq \mathcal{B}$ is *conflict free*, denoted as *cf*(S), iff $\nexists \alpha, \beta \in S$ such that $a \rightarrow \beta$. A semantics $\mathcal{S}$ satisfies the $\mathcal{CF}$ principle if and only if $\forall AF, \forall E \in \mathcal{E}_{\mathcal{S}}(AF) : cf(E)$.

Given an argumentation framework $\mathcal{AF}$, the various semantics of the argumentation framework are all based on the notion of defense. A set of arguments $\mathcal{S}$ defends an argument a when for each attacker b of a, there is an argument in $\mathcal{S}$ that attacks b. A set of acceptable arguments is called an extension. The following definition summarizes the most widely used acceptability semantics, that satisfy these two principles. Which semantics is most appropriate depends on the application domain of the argumentation theory.

Definition 3 Acceptability semantics. Let $AF = \langle \mathcal{B}, \rightarrow \rangle$ be an argumentation framework. Let $\mathcal{S} \subseteq \mathcal{B}$. $\mathcal{S}$ defends a if $\forall b \in \mathcal{B}$ such that $b \rightarrow a$, $\exists c \in \mathcal{S}$ such that $c \rightarrow b$. Let $D(\mathcal{S}) = \{a \mid \mathcal{S} \text{ defends } a\}$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{admiss}}(AF)$ iff $cf(\mathcal{S})$ and $\mathcal{S} \subseteq D(\mathcal{S})$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{compl}}(AF)$ iff $cf(\mathcal{S})$ and $\mathcal{S} = D(\mathcal{S})$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{ground}}(AF)$ iff $\mathcal{S}$ is the smallest $\mathcal{E}_{\mathrm{compl}}(AF)$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{pref}}(AF)$ iff $\mathcal{S}$ is a maximal $\mathcal{S} \in \mathcal{E}_{\mathrm{admiss}}(AF)$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{skep\text{-}pref}}(AF)$ iff $\mathcal{S} = \cap \mathcal{E}_{\mathrm{pref}}(AF)$.

$\mathcal{S} \in \mathcal{E}_{\mathbf{stable}}(AF)$ iff $cf(\mathcal{S})$ and $\forall b \in \mathcal{B} \backslash \mathcal{S} \exists a \in \mathcal{S} : a \rightarrow b$.

Our theory of argumentation is based on the following three steps:
1. Extend the set of arguments with auxiliary arguments;
2. Calculate the extensions of the extended theory using Dung's semantics;
3. For each extension of the extended theory, filter out the auxiliary arguments; the resulting sets of arguments are the extensions of the theory.

We propose to consider as arguments both single arguments, as in classical Dung's theory, and attacks between arguments, called attack arguments. In this way, we simplify the calculation of the Dung's semantics in the case of attacks of attacks, represented as usual attacks between arguments. In order to add the arguments for the attacks, we need to represent each single argument as composed by two parts, the *in* argument and the *out* argument. The *out* arguments are auxiliary arguments while the *in* ones represent the real arguments.

To define a particular argumentation framework, we have to define on the one hand how the set of arguments with auxiliary arguments is generated from the set of atomic arguments, and on the other hand which conditions argumentation frameworks have to satisfy, i.e., for which argumentation frameworks the acceptance function is defined.

3.2 Arguing about preferences among coalitions

[Modgil 2007] observes that a preference of argument a over argument b can be seen as an attack on the attack from b to a, in the sense that if a is preferred to b, then b cannot attack a. He introduces a three place attack relation, which we call here second-order attack, and [Modgil and Bench-Capon 2008] show how hierarchical second order argumentation can be represented in Dung's theory using attack arguments. This is visualized using our argumentation approach in Figure 8.a. The coalition argument D attacks the coalition argument C, but this attack is itself attacked by the stability argument B. In other words, we see each candidate coalition as an argument. Candidate coalition D attacks candidate coalition C and the stability argument B attacks this attack to set a preference between the two candidate coalitions. This is a second order attack.

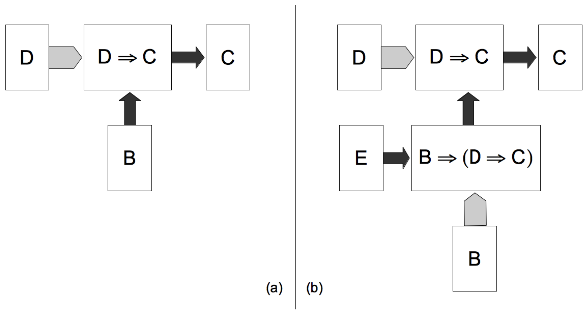

Figure 7: (a) Modgil - Bench-Capon scheme, (b) Higher order argumentation.

In Figure 7, we have two kinds of arguments, the atomic arguments and the attack arguments. We represent with the grey arrow the support relation between two arguments, e.g. argument D supports argument $D \Rightarrow C$, and with the black arrow the attack relation between two arguments, e.g. the stability argument B attacks the attack argument $D \Rightarrow C$. An argument can support another argument, e.g. an agent gives an argument which confirms a premise used by an argument provided by another agent. In our approach, the support is provided by an atomic argument for an attack argument and, usually, the supporter argument is the attacker of the attack argument to which it provides support. Attacking the attack argument $D \Rightarrow C$, the stability argument B establishes the preference of argument C over argument D.

Moreover, [Barringer et al. 2005] argue that the attack of B to $D \Rightarrow C$ can itself be attacked, where $\Rightarrow$ represents the attack. This leads to a notion of higher order attack, which we represent in our argumentation theory by Figure 8.b and Definition 4.

In coalition formation, typically coalition D and coalition C conflict, so D not only attacks C, but C also attacks D. This represents that the coalitions cannot or should not be formed together. The stability argument B represents a preference setting that coalition C is better than coalition D. Also argument E is a stability argument and it attacks the relevance of the stability argument B changing the total preference over the coalitions. At this level of abstraction, conflicts are not explicitly defined while they are described in details in the refined dynamic dependence view. Definition 4 represents our coalition view with higher order attacks.

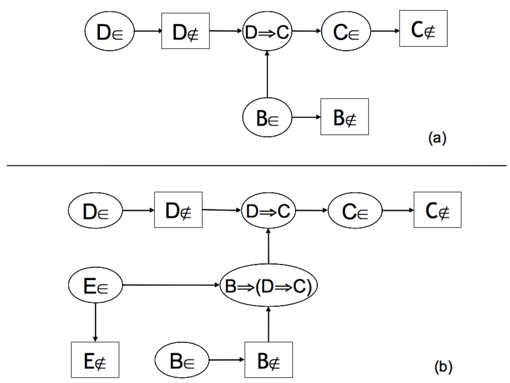

Figure 8: Expanded version of Figure 7 with *in* and *out* arguments.

Definition 4. Let A_0 be a set of atomic arguments and $a_0 \notin A_0$ a dummy argument. Let U be the minimal set of arguments such that $a_0 \in U$ and:

1. If a in A_0, $a_\in$ in U
2. If a, b in A_0, then $a \Rightarrow b$ in A_1 and in U
3. If a in A_0 and α in A_1, then $a \Rightarrow \alpha$ in A_1 and in U

For an argumentation framework $\langle \mathcal{A}, \rightarrow \rangle$, we have:

1. $a_{\in}$ in $\mathcal{A}$ iff $a \in \mathcal{A}$, and if $a_{\in}$ in $\mathcal{A}$, then $a_{\in} \rightarrow a_{\notin}$.
2. if $a \Rightarrow \alpha$ in $\mathcal{A}$ then $a, \alpha \in \mathcal{A}$, and $a_{\notin} \rightarrow (a \Rightarrow \beta)$ and $(a \Rightarrow \beta) \rightarrow \beta$.
3. there are no other attacks involved with $a_{\notin}$, and $a \Rightarrow b$ does not attack any other arguments.

Definition 4 is composed by two phases. First, the universal set of arguments is constructed, composed by the atomic arguments (1), the set of attacks between atomic arguments (2) and the set of attacks of atomic arguments to attack arguments. Second, for an argumentation framework $\langle \mathcal{A}, \rightarrow \rangle$, auxiliary arguments are added, particularly for each atomic argument we add two auxiliary arguments with the *in* argument attacking the *out* argument where $a \rightarrow b$ means the attack between every kind of arguments (1), each attack from an atomic argument to an attack argument is represented as an attack of the *out* auxiliary argument associated to the attacker to the attack argument and the attack of the attack argument to the attacked argument (2).

Figure 8 presents the expanded version of the argumentation frameworks represented in Figure 7. The main difference consists in the representation of the arguments by means of two auxiliary arguments, e.g., argument a is represented with the two auxiliary arguments $a_{\in}$ and $a_{\notin}$. These auxiliary arguments come from the Jacobovits - Vermeir - Caminada labeling [Caminada 2006] and represent whether the argument a is an element of the set of acceptable arguments or not. As a more involved example, consider the addition of attack arguments. In that case, we like to represent the attack from argument a to argument b by adding an attack argument $a \Rightarrow b$ in between. This attack argument itself attacks auxiliary argument $b_{\in}$, which represents that if the attack argument is accepted, we cannot have that argument b is accepted. Moreover, if the attack argument is not accepted, because it is itself attacked, then auxiliary argument $b_{\in}$ can be accepted, and thus argument b can be accepted. However, how do we represent that if argument a is not accepted, i.e. if the auxiliary argument $a_{\in}$ is not accepted, then we cannot have an attack from a to b either? We cannot represent this by an attack from auxiliary argument $a_{\in}$ to the attack argument, but we represent it by an attack from the auxiliary argument $a_{\notin}$ to the attack argument.This illustrates the essential role of *out* auxiliary arguments in the representation of second or higher order attacks.

Example 1 shows the application of our argumentation framework to compute which coalition is formed in each time instant using, e.g., the preferred semantics.

Example 1. Let us consider the example depicted in Figure 9. Figure 9.a represents the case of three candidate coalitions which aim to be formed in the

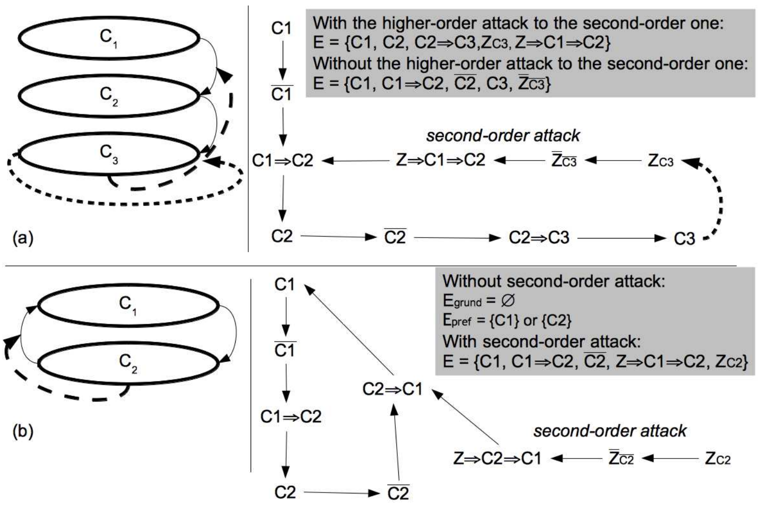

Figure 9: Candidate coalitions attacking each other from Example 1.

context of a virtual organization in a grid and this leads to the following attacks: $C_1 \Rightarrow C_2$ and $C_2 \Rightarrow C_3$. Moreover, there is also the second-order attack: $C_3 \Rightarrow (C_1 \Rightarrow C_2)$. The aim of our arguing model is to decide what coalition will be formed in this case. In Figure 9.a, candidate coalition C_3 knows that the only way to be formed consists in avoiding the formation of candidate coalition C_2. C_3 has the possibility to attack C_1's attack due to its powers, specified at the lower level of abstraction, of adding or deleting one or more of the dependencies composing C_1. C_3 decides to not use its capability of attacking the attack $C_1 \Rightarrow C_2$. The decision of C_3 of avoiding the second order attack (represented by the meta-argument Z_{C3}) in order to be formed or not is represented by means of adding an higher order attack of C_3 or not to its second order attack $C_3 \Rightarrow (C_1 \Rightarrow C_2)$. In the figure, higher order attacks are depicted as dotted arrows, while second order ones are depicted as dashed arrows on the left part and modelled as arguments in the central one. Let $\mathcal{AF} = \langle \mathcal{B}, \rightarrow \rangle$ be our argumentation framework with $C_1, C_2, C_3 \subseteq \mathcal{U}$, then the extensions of the argumentation framework, $E(\mathcal{AF})$, are as follows: if an higher order attack attacking the second order attack is added to the argumentation framework, $E(\mathcal{AF}) = \{C_1, C_3\}$, while without the higher-order attack $E(\mathcal{AF}) = \{C_1, C_2\}$. Thus, C_3 should add the higher order attack to inhibit the second order one, otherwise, C_3 will not be formed. Recall that while higher order attacks to second order attacks exiting from a coalition can be added by the attacking coalition itself to the argumentation framework, first and second order attacks are determined only by the lower levels of abstraction. Thus coalitions cannot add or delete them at their will, but they can only attack them via higher order attacks.

Figure 9.b visualizes two candidate coalitions belonging to the same grid-based virtual organization attacking each other. In this case, differently from the first one, candidate coalition C_2 does not want to be formed since, for example, it can achieve its goal without any effort if coalition C_1 is formed. Thus $C_2 \Rightarrow (C_2 \Rightarrow C_1)$. Let $\mathcal{AF} = \langle \mathcal{B}, \rightarrow \rangle$ our argumentation framework with $C_1, C_2 \subseteq \mathcal{U}$ then the extensions are, without the second order attack, $E(\mathcal{AF}) = \{C_1\}$ or $E(\mathcal{AF}) = \{C_2\}$. This situation can be seen as a sort of deadlock. Otherwise, if there is the presence of the second order attack due to the possibility for candidate coalition C_2 of adding or removing one or more of the dependencies of the concurrent coalition C_1, then the extension is $E(\mathcal{AF}) = \{C_1\}$ and the only formed coalition is C_1, as desired by coalition C_2. Figure 9.b depicts a second order attack where a stability argument, not directly represented in the figure, sets a preference of coalition C_1 over coalition C_2. According to Definition 4, there can be another stability argument setting the preference of coalition C_2 over coalition C_1, attacking by means of an higher order attack the first stability argument. This would be the case in which also coalition C_1 does not want to be formed for the same reasons of coalition C_2.

4 Analyzing reciprocity based coalitions

In this section, we answer to the third research question. First, we present the dynamic dependence view and the refined notion of coalition for this view. Second, we show how to argue on the attacks between coalitions in this refined level of abstraction.

4.1 Refining coalitions with dynamic dependencies

[Sichman and Conte 2002] introduce dependence networks, a kind of social networks representing how each agent depends on other agents to achieve the goals he cannot achieve himself. Dependence networks are based on [Castelfranchi 2003]'s basic notion of social power. They are used to specify early requirements in the TROPOS methodology [Bresciani et el. 2004], and to model and reason about agents' interactions in multiagent systems by [Sichman and Conte 2002].

Dynamic dependence networks have been introduced by [Caire et al. 2008]. In this work, a dependency between agents can depend on the interaction of other agents. Here we distinguish "negative" dynamic dependencies where a dependency exists unless it is removed by a set of agents, due to removal of a goal or ability of an agent, and "positive" dynamic dependencies where a dependency may be added due to the power of a third set of agents. As explained in the following section, these two dynamic dependencies can be used to reason

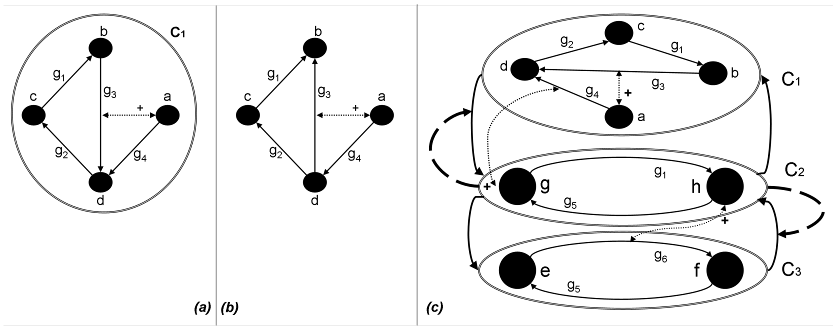

Figure 10: Agents forming a coalition (a) or not (b); the first two views (c)

about the evolution of candidate coalitions at the dynamic dependence view level of abstraction.

Definition 5 Dynamic Dependence View. A dynamic dependence network is a tuple $\langle A, G, T, \mathit{dyndep}^-, \mathit{dyndep}^+, \geq \rangle$ where:

- A is a set of agents, G is a set of goals and T is a set of time instants.
- $\mathit{dyndep}^- : A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of an agent and two sets of agents all the sets of goals in which the first depends on the second, unless the third deletes the dependency.
- $\mathit{dyndep}^+ : A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of an agent and two sets of agents all the sets of goals on which the first depends on the second, if the third creates the dependency.
- $\geq: A \rightarrow 2^G \times 2^G$ is a total pre-order on goals which occur in each agent's dependencies: $G_1 \geq (a) G_2$ implies that $\exists B, C \subseteq A$ such that $a \in B$ and $G_1, G_2 \in \mathit{dyndep}^-(a, B, C)$ or $G_1, G_2 \in \mathit{dyndep}^+(a, B, C)$.

The static dependencies are defined by $\mathit{dep}(a, B) = \mathit{dyndep}^-(a, B, \emptyset)$.

Example 2. Let us consider the example depicted in Figure 10.a where we have four nodes belonging to a grid-based virtual organization. Node b depends on node d for goal g_3, if node a creates this dependency: $\mathit{dep}(a, \{d\}) = \{\{g_4\}\}$, $\mathit{dep}(d, \{c\}) = \{\{g_2\}\}$, $\mathit{dep}(c, \{b\}) = \{\{g_1\}\}$, $\mathit{dyndep}^+(b, \{d\}, \{a\}) = \{\{g_3\}\}$.

A coalition can be defined in dependence networks, based on the idea that to be part of a coalition, every agent has to contribute something, and has to get something out of it. Roughly, a coalition can be formed when there is a cycle of dependencies (the definition of coalitions is more complicated due to the fact that an agent can depend on a set of agents, see below). We show how dependence networks can be used for coalition evolution, by assuming that goals are maintenance goals rather than achievement goals, which give us automatically a longer term and more dynamic perspective.

We define reciprocity based coalitions for dynamic dependence networks, firstly introduced by [Boella et al. 2008], as a refinement of the coalition view. We represent the coalition not only as a set of agents, like in game theoretical approaches, but as a set of agents together with a partial dynamic dependence relation. Intuitively, the dynamic dependence relation represents the "contract" of the coalition: if $H \in dyndep^+(a, B, D)$, then the set of agents D is committed to create the dependency, and the set of agents B is committed to see to the goals H of agent a. The rationality constraint on such reciprocity based coalitions is that each agent contributes something, and receives something back. Our notion of coalition presents the agents composing it not only as utility maximizers as in coalitional game theoretical approaches but as complex entities with their sets of beliefs and goals which have to be satisfied. In our approach, coalitions have complex structure, composed by existing dependencies and potential ones which represent a kind of dynamic contract.

Definition 6 Reciprocity based Coalition. Given a dynamic dependence network $\langle A, G, T, dyndep^-, dyndep^+, \geq \rangle$, a reciprocity based coalition is represented by coalition $C \subseteq A$ together with dynamic dependencies $dyndep^{+'} \subseteq dyndep^+$, such that:

– if $\exists b, B, D, H$ with $H \in dyndep^{+'}(a, B, D)$ then $a \in C$, $B \subseteq C$ and $D \subseteq C$ (the domain of $dyndep^{+'}$ contains only agents in coalition C), and

– for each agent $a \in C$ we have $\exists b, B, D, H$ with $H \in dyndep^{+'}(b, B, D)$ such that $a \in B \cup D$ (agent a contributes something, either creating a dependency or fulfilling a goal), and

– for each agent $a \in C$ $\exists B, D, H$ with $H \in dyndep^+(a, B, D)$ (agent a receives something from the coalition).

The following example illustrates that dependencies will be created by agents only if these new dependencies work out in their advantage.

Example 3 Continued. Each agent of $C_1 = \{a, b, c, d\}$ creates a dependency or fulfills a goal. Figure 10.a represents a set of agents composing a coalition in accordance with Definition 6 while Figure 10.b represents the same set of agents not forming a coalition. The difference among the two figures consists in the direction of the arrow joining agents b and d.

4.2 Maintaining or destroying coalitions

The basic attack relations between coalitions are due to the fact that coalitions are based on the same goals, differently from the conflicts between coalitions in Amgoud's coalition theory [Amgoud 2005] where two coalitions are based on the same tasks. In the coalition view, we distinguish between two kinds of attacks: first order ones, between atomic arguments, and higher order attacks, between atomic arguments and attack arguments. In the dynamic dependence view, we details these two kinds of attacks. Attack relations between coalitions sharing the same goals are the refined version of first order attacks presented in the coalition view. In this view, we present first order attacks as the reciprocal attacks between coalitions without coming into details of the reasons behind these attacks. In this refined view, the reason is characterized by the sharing of a goal between the two (or more) coalitions. In this case, the two candidate coalitions cannot be formed together since an agent cannot be part of two coalitions at the same time, particularly if the two candidate coalitions are based on the same goal since each goal cannot be achieved concurrently by more than one agent.

Definition 7 First order attack. Coalition $\langle C_1, dyndep_1\rangle$ attacks coalition $\langle C_2, dyndep_2\rangle$ if and only if there exist $a_1, a_2, B_1, B_2, D_1, D_2, G_1, G_2$ such that $G_1 \in dyndep_1(a_1, B_1, D_1)$, $G_2 \in dyndep_2(a_2, B_2, G_2)$ and $G_1 \cap G_2 \neq \emptyset$.

Figure 11 aims to represent in the refined version the two cases in which a coalition wants or not to be formed. In Figure 11 are depicted two candidate coalitions composed by 3 nodes of the grid. On the one hand, in the first case we have that both the two candidate coalitions want to be formed. This is a sort of deadlock situation but it would be solved thanks to the presence of eventual dynamic dependencies. These two candidate coalitions are attacking each other as the first two coalitions of Figure 9.a. On the other hand, in the second case we have that both nodes a and c depend on node b to run the file results.mat and both of them know that if the other coalition is formed goal g_1 will be achieved without any effort. These two candidate coalitions are attacking each other but if, for example, coalition C_2 has the possibility to delete one of its dependencies then this higher order attack would decide the formation of coalition C_1. In this way, coalition C_2 has obtained its aim and goal g_1 will be achieved by agent a.

Definition 8 presents three different classes in which we divide the set of candidate coalitions due to their features and the sign, positive or negative, of the dynamic dependencies composing them.

Definition 8. Let A be a set of agents and G be a set of goals. A coalition function is a partial function $C : A \times 2^A \times 2^G$ such that $\{a \mid C(a, B, G)\} = \{b \mid b \in B, C(a, B, G)\}$, the set of agents profiting from the coalition is the set of agents contributing to it. Let $\langle A, G, T, dyndep^-, dyndep^+, \geq\rangle$ be a dynamic dependence network, and dep the associated static dependencies.

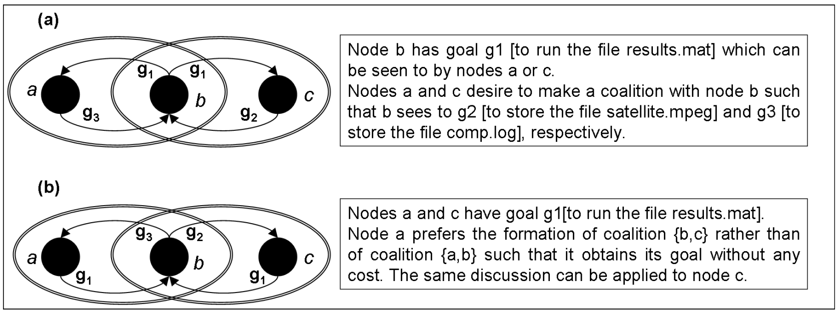

Figure 11: Candidate coalitions sharing goals.

1. A coalition function C is a coalition if $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in dep(a,B)$. These coalitions which cannot be destroyed by addition or deletion of dependencies by agents in other coalitions.

2. A coalition function C is a vulnerable coalition if it is not a coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in \cup_D dyndep^-(a,B,D)$. Coalitions which do not need new goals or abilities, but whose existence can be destroyed by removing dependencies.

3. A coalition function C is a potential coalition if it is not a coalition or a vulnerable coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in \cup_D (dyndep^-(a,B,D) \cup G' \in dyndep^+(a,B,D))$. Coalitions which could be created or which could evolve if new abilities or goals would be created by agents of other coalitions on which they dynamically depend.

There are various further refinements of the notion of coalition. For example, [Boella et al. 2006] look for minimal coalitions. In this paper we do not consider these further refinements.

Higher order attacks are detailed in the dynamic dependence view by removing or adding one of the dependencies of the attacked coalition. This kind of attack is the refined version of higher order attacks of the coalition view and is represented by means of the stability argument. This kind of attack relation means a real destruction of the attacked coalition since one or more of its dependencies are deleted or added and the coalition does not exist any more. The stability argument establishes the preference and the preferred coalition is preserved by these additions and removals and thus it maintains its stability. Higher order attacks attack the coalition by attacking all its attacks to the other coalitions, independently from what coalition is doing the higher order attack.

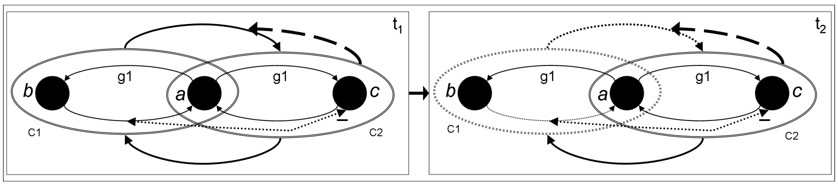

Figure 12: Two vulnerable coalitions attacking each other.

Definition 9 Higher order attack. $\forall C_1, C_2$ such that $C_1 \Rightarrow C_2$, coalition $\langle C, dyndep\rangle$ attacks the attack from coalition $\langle C_1, dyndep_1\rangle$ on coalition $\langle C_2, dyndep_2\rangle$ if and only if there exists a set of agents $D \subseteq \{a \mid \exists E, H, C(a, E, H)\}$ such that $\exists a, B, G', C_1(a, B, G')$ and $G \in dyndep(a, B, D)$.

Higher order attacks, presented in definition 9, can arise if the coalition C which has to attack the attack $C_1 \Rightarrow C_2$ is composed by a set of agents D such that they can add or delete at least one dynamic dependency.

Example 4. Assume we have eight agents, $a, ..., h$ and the dependencies of Example 2, depicted in Figure 10.c: $dep(a, \{d\}) = \{\{g_4\}\}$, $dep(d, \{c\}) = \{\{g_2\}\}$, $dep(c, \{b\}) = \{\{g_1\}\}$, $dyndep^+(b, \{d\}, \{a\}) = \{\{g_3\}\}$, plus the following ones: $dep(e, \{f\}) = \{\{g_6\}\}$, $dep(f, \{e\}) = \{\{g_5\}\}$, $dep(g, \{h\}) = \{\{g_1\}\}$, $dep(h, \{g\}) = \{\{g_5\}\}$, $dep(c, \{h\}) = \{\{g_1\}\}$, $dep(g, \{b\}) = \{\{g_1\}\}$, $dep(h, \{e\}) = \{\{g_5\}\}$, $dep(f, \{g\}) = \{\{g_5\}\}$.

The possible coalitions are C_1, C_2 and C_3 where:

$C_1 = \{dep(a, \{d\}) = \{\{g_4\}\}, dep(d, \{c\}) = \{\{g_2\}\}, dep(c, \{b\}) = \{\{g_1\}\},$
$dyndep^+(b, \{d\}, \{a\}) = \{\{g_3\}\}\},$
$C_2 = \{dep(e, \{f\}) = \{\{g_6\}\}, dep(f, \{e\}) = \{\{g_5\}\}\},$
$C_3 = \{dep(g, \{h\}) = \{\{g_1\}\}, dep(h, \{g\}) = \{\{g_5\}\}\}.$

Some of the dependencies remain outside all coalitions (e.g., $dep(c, \{h\}) = \{\{g_1\}\}$, $dep(g, \{b\}) = \{\{g_1\}\}$, $dep(h, \{e\}) = \{\{g_5\}\}$, $dep(f, \{g\}) = \{\{g_5\}\}$, not reported in Figure 10.c). Thus, $C_1 \Rightarrow C_2$, $C_2 \Rightarrow C_1$, $C_2 \Rightarrow C_3$ and $C_3 \Rightarrow C_2$ due to the fact that they share goals g_1 and g_5 respectively. Note that these attacks are reciprocal. The coalitions attack each other since agents b and h on which respectively c and g depend for g_1 would not make their part hoping that the other one will do that, so to have a free ride and get respectively g_3 achieved by d and g_5 by g.

Figure 12 illustrates a new example of conflict among vulnerable coalitions.

Example 5. Using the grid-based scenario, we can model the example depicted in Figure 12. Assume, in the first time instant t_1, we have a portion of a virtual organization composed by three nodes, a, b, c represented as agents in our model. There are three goals g_1: to run the file *results.mat*, g_2: to save the file *satellite.mpeg*, g_3: to save the file *comp.log*.

These goals, associated to the power of the agents to achieve them, form the following dependencies among the agents (we write $C(a, b, g_1)$ for $C(a, \{b\}, \{g_1\})$ and $dep(a, b, g_1)$ for $dep(a, \{b\}) = \{g_1\}$): $dep(a, b, g_1)$, $dep(a, c, g_1)$, $dep(b, a, g_2)$, $dep(c, a, g_3)$. The situation is that node a depends on both nodes b and c to run the file *results.mat* and thus to obtain the results of his job, node b depends on node a for the storage of file *satellite.mpeg* and, finally, node c depends on node a for the storage of file *comp.log*. Thus, there are two candidate coalitions: $C_1 = \{(a, b, g_1), (b, a, g_2)\}, C_2 = \{(a, c, g_1), (c, a, g_3)\}$.

They will not create both since one is enough for node a to have someone look after his goal g_1: $C_1 \Rightarrow C_2$ and $C_2 \Rightarrow C_1$. Now, we assume that node c removes the necessity of node b to store the file *satellite.mpeg*, destroying the dependency $dep(b, a, g_2)$, i.e., we substitute it with $dyndep^-(b, a, c, g_2)$, e.g., by removing the power of node a to see to goal g_2, or by removing the goal g_2 of node b. This deletion, shown in time instant t_2 of Figure 12, allows node c to ensure himself the dependency on himself of node a to perform his job, goal g_1. In this way, node c ensures himself the help of node a to store file *comp.log*. This deletion sets a preference relation of the candidate coalition C_2, represented here with the attack of coalition C_2 to the attack relation of coalition C_1 to coalition C_2. In this case, coalition $C_2 = \{(a, c, g_1), (c, a, g_3)\}$ will become the only possible extension, since $C_2 \Rightarrow (C_1 \Rightarrow C_2)$ by Definition 9.

5 Future research

In this section, we refine the abstract coalition models with powers of sets of agents and the conditional goals of the agents. We present two more refined viewpoints, the power view and the agent view, but the analysis of reciprocity based coalitions at these refined levels is not provided in the paper and it is left for future research. In classical planners, goals are unconditional. Therefore, many models of goal based reasoners, including the model of [Boella et al. 2004], define the goals of a set of agents A by a function $goals : A \rightarrow 2^G$, where G is the complete set of goals. However, in many agent programming languages and architectures, goals are conditional and can be generated. The power to trigger a goal is distinguished from the power to fulfill a goal.

Definition 10 Power view. The Power view is represented by the tuple $\langle A, G, X, T, goals, power\text{-}goals, power\rangle$, where A, G, X and T are sets of agents, goals, decision variables, and time instants, $goals : A \rightarrow 2^G$, and $power\text{-}goals : 2^A \rightarrow 2^{(A \times G)}$ is a function associating with each set of agents the goals they can create for agents, and $power : 2^A \rightarrow 2^G$ is a function associating with agents the goals they can achieve.

The function power represents what goals each agent or group of agents can achieve without being supported by other agents. For example, $power(\{a_1\}) = \{g_1\}$ means that agent a_1 is able to achieve g_1. Note that it is not given that g_1 is a goal of agent a_1. We therefore extend the agent view with conditional goals.

Definition 11 Agent view. The Agent view is represented by the tuple $\langle A, G, X, T, goals, skills, R\rangle$, where A, G, X, T are disjoint sets of agents, goals, decision variables, and time instants, *goals* is as before, $skills : A \rightarrow 2^X$ is a function associating with an agent its possible decisions, and $R : 2^X \rightarrow 2^G$ is a function associating with decisions the goals they achieve.

The power view can be defined as an abstraction of the agent view. A set of agents B has the power to see to it that agent a has the goal g, written as $(a, g) \in power\text{-}goals(B)$, if and only if there is a set of decisions of B such that g becomes a goal of a. A set of agents B has the power to see to goal g if and only if there is a set of decisions of B such that g is a consequence of it.

Definition 12. $\langle A, G, T, goals, power\text{-}goals, power\rangle$ is an abstraction from $\langle A, G, X, T, goals, skills, R\rangle$ if and only if: $(a, g) \in power\text{-}goals(B)$ if and only if $\exists Y \subseteq skills(B)$ with $skills(B) = \cup\{skills(b) \mid b \in B\}$ such that $g \in goals(a, Y)$, and $g \in power(B)$ if and only if $\exists Y \subseteq skills(B)$ such that $g \in R(Y)$.

Abstracting power view to a dynamic dependence network can be done as follows. Note that in this abstraction, the creation of a dynamic dependency is based only on the power to create goals. In other models, creating a dependency can also be due to creation of new skills of agents.

Definition 13. $\langle A, G, T, dyndep^-, dyndep^+, \geq\rangle$ is an abstraction of $\langle A, G, T, power\text{-}goals, power\rangle$, if we have $H \in dyndep^+(a, B, C)$ if and only if $\forall g \in H : (a, g) \in power\text{-}goals(C)$, and $H \subseteq power(B)$.

Combining these two abstractions, abstracting agent view to a dynamic dependence view can be done as follows.

Proposition 14. *$\langle A, G, T, dyndep^-, dyndep^+, \geq\rangle$ is an abstraction of $\langle A, G, X, T, goals, skills, R\rangle$, if we have $H \in dyndep^+(a, B, C)$ if and only if $\exists Y \subseteq skills(C)$ such that $H \subseteq goals(a, Y)$, and $\exists Y \subseteq skills(B)$ such that $H \subseteq R(Y)$.*

The arguing at these refined levels of abstraction is our main aim for future work. The approach we plan to apply will follow the examples provided in [Boella et al. 2005] and [Amgoud and Prade 2009], particularly concerning the agent view in which we describe an agent by means of the same features of these works such as goals, beliefs and so on. The main difference concerns the power view which is not considered in these works and which has to be representing taking into account also the implicit notion of group present in it.

6 Related Work

This paper is a revised and extended version of the papers [Boella et al. 2008] and [Boella et al. 2008], where in the former we introduced the four viewpoints and in the latter we introduced higher order attacks, while in this paper we add iterative design.

Although there were many approaches defining coalition formation, two represents different perspectives: the model of [Shehory and Kraus 1998] and the one of [Sichman 1998]. [Shehory and Kraus 1998] present algorithms that enable the agents to form groups and assign a task to each group, calling these groups coalitions. [Sichman 1998] presents coalition formation using a dependence-based approach based on the notion of social dependence [Castelfranchi 2003]. Another definition of coalition inspired by dependence networks is given by [Boella et al. 2006]. See [Sauro 2005] for a further discussion. Once represented the internal structure of coalitions, one could study which kind of relations there are among candidate coalitions at an higher level of detail disregarding which are the causes for incompatibility. In this paper we use an argument labeling by [Jakobovits and Vermeir 1999] and [Caminada 2006]. They show that semantics can also be described by a three valued argument labeling, where the first two conditions represent conflict free and defense, and the third one represents the so-called reinstatement principle. They show how the other semantics can be defined in terms of these labelings too. The generation of arguments with auxiliary arguments and the condition on argumentation frameworks are formalized by [Modgil 2007]. For a further discussion, see [Boella et al. 2008].

The application of argumentation frameworks to coalition formation has been discussed by [Amgoud 2005] and by [Bulling et al. 2008]. The latter combines the argumentation framework and ATL presenting a generalization of Dung's argumentation framework, extended with a preference relation. Alternating-time temporal logic is a temporal logic that is used for reasoning about the behavior and abilities of agents under various rationality assumptions. The key construct in ATL expresses that a coalition of agents can enforce a given formula [Alur et al. 2002]. [Amgoud 2005], instead, proposes to use an extension of Dung's argumentation theory with preferences and associated dialogue theories as a formal framework for coalition formation. As preferred extensions exist for every argumentation framework, we can introduce the preferred solutions to coalitional games by defining them as the preferred extensions of the corresponding argumentation system. Amgoud illustrates this idea by formalizing a task based theory of coalition formation, where the conflict relation represents that two coalitions contain the same task. However, a drawback of this abstract approach is that it is less clear where the preferences among coalitions come from. In contrast with our approach, a coalition is viewed as an abstract entity whose structure is not known. Unlike Amgoud's work, we do not provide this paper with a proof theory since it is derivable from the argumentation theory's literature. Another formal approach to reason about coalitions is coalition logic [Pauly 2002] and ATL [Alur et al. 2002], describing how a group of agents can achieve a set of goals, but without considering the internal structure of the group of agents [van der Hoek et al. 2005].

Concerning design, the TROPOS methodology [Bresciani et el. 2004] covers five phases of the software development process: the early requirements allowing to analyse and model the requirements of the context in which the software system will be inserted, late requirements describing the requirements of the software system, architectural design and detailed design aiming to design the architecture of the system and, finally, the code implementation. The idea of focusing the activities that precede the specification of software requirements, in order to understand how the intended system will meet organizational goals, has been first proposed in requirements engineering, specifically in [Yu 1995] 's work with his i* model . The i* model offers actors, goals and actor dependencies as primitive concepts. The rationale of the i* model is that by doing an earlier analysis, one can capture not only the what or the how, but also the why a piece of software is developed. This, in turn, supports a more refined analysis of system dependencies and encourages a uniform treatment of the system's functional and non-functional requirements.

7 Summary

In this paper, we present an approach to iteratively design social networks by introducing four viewpoints, the refinement relations between them, and the methods to analyze cooperation based on emerging coalitions. Iterative design is a design methodology based on a cyclic process of analyzing and refining a work in progress. In iterative design, interaction with the designed system is used as a form of research for evolving a project and successive iterations of a design are implemented. The designer starts with the more abstract level and refines it in each step to the level below it.

We analyze the reciprocity based coalitions that emerge in social networks at various degrees of abstraction. At the most abstract viewpoint, coalitions are abstract entities and we adapt existing coalition argumentation theory to reason about these coalitions seen as arguments. We introduce the stability argument preferring a coalition over the others and the attack relations between them. This argumentation theory allows to model the attacks among candidate coalitions and to decide if a coalition is formed. This analysis is refined in the dynamic dependence view providing the composition of each coalitions and the reasons behind the attacks and the preferences between them. Further analysis at the most refined views is left for future work.

We refine abstract coalition models with social dynamic dependencies among agents, powers of sets of agents, and plans by making the dependence relation conditional to the agents that have the power to create or delete it. These dynamic dependencies are higher order dependencies reflecting the behaviours of the more abstract higher order attacks of the coalition view. A further refinement leads to the definition the power view and the agent view. The agent view is the most detailed view considering all the features of the single agents as facts, goals and skills but it looses the notion of "group" which is present, instead, in the power view, associating a *set of* agents to the goals they can achieve.

Subjects of further research are the use of our new theory for coalition formation. For example, when two agents can make the other depend on itself and thus create a potential coalition, when will they do so? Moreover, in this paper we concentrate our attention on single coalitions. We aim to extend this model by considering more than one formed coalition which cooperates with other coalitions in order to achieve an increased outcome. From this point of view, the extended model represent each coalition as a node of an argumentation network in which coalitions have to manage attack decisions and coalitions can aggregate to each other due to their decisions and the achievable outcome represented by a game.

References

[Alur et al. 2002] Alur, R. Henzinger, T. Kupferman, O: "Alternating-time temporal logic"; Journal of ACM, 49(5), 672-713, (2002).

[Amgoud 2005] Amgoud, L.: "An Argumentation-Based Model for Reasoning About Coalition Structures"; Proc. 2^{nd} Int. Workshop on Argumentation in Multi-Agent Systems, LNCS, Springer, 217-228, (2005).

[Amgoud and Prade 2009] Amgoud, L. Prade, H.: "Using arguments for making and explaining decisions"; Artif. Intell. 173 (3-4), 413-436, (2009).

[Baroni and Giacomin 2007] Baroni, P. Giacomin, M.: "On principle-based evaluation of extension-based argumentation semantics"; Artif. Intell. 171, 675-700, (2007).

[Barringer et al. 2005] Barringer, H. Gabbay, D. M. Woods, J.: "Temporal Dynamics of Support and Attack Networks: From Argumentation to Zoology"; Mechanizing Mathematical Reasoning, 59-98, (2005).

[Bench-Capon and Dunne 2007] Bench-Capon, T. J. M. Dunne, P. E.: "Argumentation in artificial intelligence"; Artif. Intell. 171(10-15): 619-641, (2007).

[Boella et al. 2004] Boella, G. Sauro, L. van der Torre, L.: "Social Viewpoints on multiagent Systems"; Proc. 3^{rd} Int. Joint Conf. on Autonomous Agents and Multiagent Systems, 1358-1359, (2004).

[Boella et al. 2005] Boella, G. Hulstijn, J. van der Torre, L.: "Argumentation for Access Control"; Proc. 9^{th} Congress of the Italian Association for Artificial Intelligence, 86-97, (2005).

[Boella et al. 2006] Boella, G. Sauro, L. van der Torre, L.: "Strengthening Admissible Coalitions"; Proc. 17^{th} European Conf. on Artificial Intelligence, 195-199, (2006).

[Boella et al. 2008] Boella, G. van der Torre, L. Villata, S.: "Social Viewpoints for Arguing about Coalitions"; Proc. 11^{th} Pacific Rim Int. Conf. on Multi-Agents, LNCS, Springer, 66-77, (2008).

[Boella et al. 2008] Boella, G. van der Torre, L. Villata, S.: "Attack Relations among Dynamic Coalitions"; Proc. 20^{th} Belgian-Netherlands Conference on Artificial Intelligence, 25-32, (2008).

[Bresciani et el. 2004] Bresciani, P. Perini, A. Giorgini, P. Giunchiglia, F. Mylopoulos, J.: "Tropos: An Agent-Oriented Software Development Methodology"; Autonomous Agents and Multi-Agent Systems Journal, 8, 203-236, (2004).

[Bulling et al. 2008] Bulling, N. Chesnevar, C. I. Dix, J.: "Modelling Coalitions: ATL + Argumentation"; Proc. 7^{th} Int. Conf. on Autonomous Agents and Multiagent Systems, 681-688, (2008).

[Caire et al. 2008] Caire, P. Villata, S. van der Torre, L. Boella, G.: "Conviviality masks in multiagent systems"; Proc. 7^{th} Int. Conf. on Autonomous Agents and Multiagent Systems, 1265-1268, (2008).

[Caminada 2006] Caminada, M.: "On the Issue of Reinstatement in Argumentation"; Proc. 10^{th} European Conf. on Logics in Artificial Intelligence, 111-123, (2006).

[Castelfranchi 2003] Castelfranchi, C.: "The micro-macro constitution of power, Protosociology", 18, 208-269, (2003).

[Dung 1995] Dung, P. M.: "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games", Artif. Intell. 77(2), 321-357, (1995).

[Foster et al. 2001] Foster, I. Kesselman, C. Tuecke, S.: "The Anatomy of the grid: Enabling Scalable Virtual Organizations"; Int. J. of Supercomputer Applications, 15, (2001).

[Hanneman and Riddle 2005] Hanneman, R. Riddle, M.: "Introduction to social network methods", (2005).

[Jakobovits and Vermeir 1999] Jakobovits, H. Vermeir, D.: "Robust Semantics for Argumentation Frameworks", J. Log. Comput. 9(2), 215-261, (1999).

[Modgil 2007] Modgil, S.: "An Abstract Theory of Argumentation That Accommodates Defeasible Reasoning About Preferences", Proc. 9^{th} European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 648-659, (2007).

[Modgil and Bench-Capon 2008] Modgil, S. Bench-Capon, T.: "Integrating Object and Meta-Level Value Based Argumentation"; Proc. 2^{nd} Int. Conf. on Computational Models of Argument, 172, 240-251, (2008).

[Pauly 2002] Pauly, M.: "A Modal Logic for Coalitional Power in Games"; J. of Logic and Computation, 12, 146-166, (2002).

[Sauro 2005] Sauro, L.: "Formalizing admissibility criteria in coalition formation among goal directed agents", University of Turin, PhD Thesis, (2005).

[Sichman 1998] Sichman, J. S.: "DEPINT: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario"; J. of Artificial Societies and Social Simulation 1(2), (1998).

[Sichman and Conte 2002] Sichman, J. S. Conte, R.: "Multi-agent dependence by dependence graphs"; Proc. 1^{st} Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, 483-490, (2002).

[Shehory and Kraus 1998] Shehory, O. Kraus, S.: "Methods for task allocation via agent coalition formation"; Artif. Intell. 101, 165-200, (1998).

[van der Hoek et al. 2005] van der Hoek, W. Jamroga, W. Wooldridge, M.: "A Logic for Strategic Reasoning", Proc. 4^{th} Int. Joint Conf. on Autonomous Agents and Multi Agent Systems, 157-164, (2005).

[van der Torre and Villata 2009] van der Torre, L. Villata, S.: "Four Ways to Change Coalitions: Agents, Dependencies, Norms and Internal Dynamics", Proc. 1^{st} Symposium on Social Networks and Multiagent Systems, AISB 2009 Convention, 19-24, (2009).

[Yu 1995] Yu, E.: "Modelling Strategic Relationships for Process Reengineering", University of Toronto, PhD Thesis, (1995).