

# Computing coalitions in Multiagent Systems. A contextual reasoning approach

Antonios Bikakis<sup>1</sup> and Patrice Caire<sup>2</sup>

<sup>1</sup> Department of Information Studies, UCL, UK  
a.bikakis@ucl.ac.uk

<sup>2</sup> SnT and CSC, University of Luxembourg, Luxembourg  
patrice.caire@uni.lu

**Abstract.** In multiagent systems, agents often have to rely on other agents to reach their goals, for example when they lack a needed resource or do not have the capability to perform a required action. Agents therefore need to cooperate. Some of the questions then raised, such as, which agent to cooperate with, are addressed in the field of coalition formation. In this paper we go further and first, address the question of how to compute the solution space for the formation of coalitions using a contextual reasoning approach. We model agents as contexts in Multi-Context Systems (MCS) and dependence relations among agents as bridge rules. We then systematically compute all potential coalitions using algorithms for MCS equilibria. Finally, given a set of functional and non-functional requirements, we propose ways to select the best solutions. We illustrate our approach with an example from robotics.

## 1 Introduction

In multiagent systems, agents have goals to satisfy. Typically, agents cannot reach all their goals by themselves, without any help. Instead, agents need to cooperate with other agents, for example because they need a specific resource to satisfy a goal, or do not have the capability required to perform a task.

The questions then, are: Which agent to cooperate with? Which group of agents to join? The problem of assembling a group of cooperating agents in order for all agents to reach their goals, shared or not, is referred to as coalition formation, and has been on the focus of many recent works in the area of multiagent systems (e.g., [31, 32, 30, 18, 3, 17, 7]). This paper introduces a novel contextual reasoning approach to address the problem based on the use of Multi-Context Systems (MCS).

*Multi-Context Systems (MCS)* [16, 15, 6] are logical formalizations of distributed context theories connected through a set of bridge rules, which enable information flow between different contexts. A *context* can be thought of as a logical theory - a set of axioms and inference rules - that models local knowledge of an agent. Intuitively, MCS can be used to represent any information system that consists of heterogeneous knowledge agents including peer-to-peer systems, distributed ontologies or Ambient Intelligence systems. Several applications have

already been developed on top of MCS or other similar formal models of context including (a) CYC common sense knowledge base [21], (b) contextualized ontology languages, such as Distributed Description Logics [4] and C-OWL [5], (c) context-based agent architectures [25, 26], and (d) distributed reasoning algorithms for Mobile Social Networks [1] and Ambient Intelligence systems [2].

Here we address the question of how to find and evaluate coalitions among agents while taking advantage of the MCS model and algorithms. The main advantages of this approach are: (a) MCS can represent heterogenous multiagent systems, i.e. systems containing agents with different knowledge representation models; (b) bridge rules can represent different kinds of inter-agent relationships such as dependencies, constraints and conflicting goals; (c) there are both centralized and distributed algorithms that can be used for computing the potential coalitions. We formulate our main research question as:

- *How to find and evaluate coalitions among agents in multiagent systems using MCS tools?*

This breaks down into the following two sub-questions:

1. *How to formally compute the solution space for coalition formation using MCS tools?*
2. *How to select the best solution given a set of requirements?*

Our methodology is the following. We start with modeling dependencies among agents using dependence relations as described in [32]. We then model the system as a MCS: each agent is modeled as a context with a knowledge base with an underlying logic and dependence relations are modeled as bridge rules. Third, we use appropriate algorithms to compute MCS equilibria. Each equilibrium corresponds to a different coalition. Finally, given a set of requirements, we show how to select the best solutions. The requirements we consider may be of two kinds. They may be domain related. For example in robotics, power consumption is a key concern that must be carefully dealt with. They may also be system related. For example in multiagent systems, the efficiency and conviviality of the system may be considered.

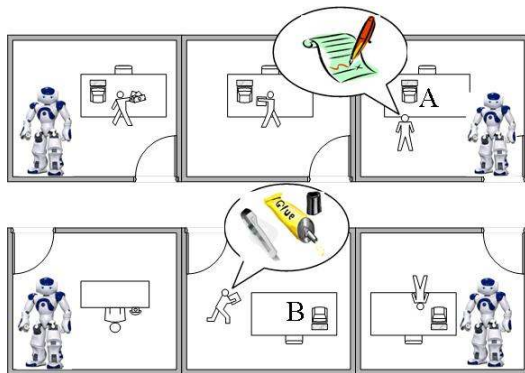
The rest of the paper is structured as follows. Section 2 introduces our running example from robotics. Section 3 presents background information on dependence networks, coalition formation and MCS. Section 4 describes our approach: how we use MCS to represent agents and their dependencies; how we systematically compute the coalitions; and how we then select the best coalitions with respect to given requirements. Section 5 presents related research, and Section 6 concludes with a summary and a perspective on future works.

## 2 Running Example

We now present a scenario to illustrate how our approach works. Consider an office building, where robots assist human workers. As typically, there are not enough office supplies, such as cutters, glue, etc., for everyone, they have to be

shared among the workers. Furthermore, as it is considered inefficient and unproductive for a worker to contact other colleagues and get supplies by themselves, the worker can submit a request to the robots to get and/or deliver the needed supplies for her, while she/he keeps on working at her desk. We refer to a request submitted to the robots as a task.

Workers and robots communicate via a simple web-based application, which transmits the workers' requests to the robots and keeps track of their status. The robots have limited computational resources: they only keep track of their recent past. Furthermore, not all robots know about the exact locations of supplies. Therefore, robots rely on each other for information about the location of the supplies: the last robot having dealt with a supply is the one knowing where it is. We assume the availability of such an application, and a stable and reliable communication network. A depiction of the scenario is presented in Figure 1.



**Fig. 1.** A depicted scenario of robots in office building.

We consider a set of 4 robots  $A = \{ag_1, ag_2, ag_3, ag_4\}$  and four tasks:  $T = \{t_1, t_2, t_3, t_4\}$ , where:  $t_1$  is to deliver a pen to desk A,  $t_2$  is to deliver a piece of paper to desk A,  $t_3$  is to deliver a tube of glue to desk B, and  $t_4$  is to deliver a cutter to desk B. We assume that a robot can perform a task if it can carry the relevant supply and knows its source and destination. Due to their functionalities, robots can carry the following supplies:  $ag_1$  the pen or the glue,  $ag_2$  the paper,  $ag_3$  the glue or the cutter, and  $ag_4$  the pen or the cutter. Each robot knows who has the information about the source and the destination of each supply, but the actual coordinates are only revealed after an agreement on a coalition among the robots has been made. This involves interdependency among robots.

To start, robots get the information concerning the locations of the supplies and the distances between the supplies and their destinations. Tables 1 and 2 present the knowledge of the robots about the tasks and the current distances among the robots, the supplies and the destinations, respectively. The table should be read as follows. Robot  $ag_1$ , regarding task  $t_1$ , knows nothing about the source of the pen, i.e., where it currently is, but does know the destination

for the pen, i.e., where it must be delivered. Regarding task  $t_2$ , robot  $ag_1$  knows where the paper is, but knows nothing about its destination.

Upon receiving information about the tasks, robots generate plans to carry out the tasks based on the knowledge and the capabilities of each robot. For example, there are two different plans for delivering the pen to desk A ) $t_1$ :  $ag_1$  can deliver it after receiving information about its location from robot  $ag_2$ ; alternatively,  $ag_4$  can deliver it after receiving information about its location from  $ag_2$  and about its destination from  $ag_1$ . Given the plans, the robots then need to decide how to form coalitions to execute the tasks. We refer to a coalition as a group of robots executing a task. For example to accomplish all tasks  $t_1, t_2, t_3, t_4$ , the following coalitions may be formed:  $C_0 : \{(ag_1, t_3), (ag_2, t_2), (ag_3, t_4), (ag_4, t_1)\}$  and  $C_1 : \{(ag_1, t_1), (ag_2, t_2), (ag_3, t_3), (ag_4, t_4)\}$ .

After forming coalitions, each robot has to generate its own plan to carry out the assigned tasks, e.g. plan the optimal route to get the supply and carry it to its destination. Optimal route planning is a typical shortest path finding algorithm, i.e., implementations are available and can be deployed on the robots. Therefore, the robots can generate plans for themselves after they have been given tasks. Details about generating plans for the robots is out of the scope of the paper.

Robot	$ag_1$				$ag_2$			
Task	$t_1$	$t_2$	$t_3$	$t_4$	$t_1$	$t_2$	$t_3$	$t_4$
Source		x				x		
Destination	x		x					x

Robot	$ag_3$				$ag_4$			
Task	$t_1$	$t_2$	$t_3$	$t_4$	$t_1$	$t_2$	$t_3$	$t_4$
Source				x			x	
Destination		x						

**Table 1.** Robots’ knowledge and capabilities

Robot	Pen	Paper	Glue	Cutter
$ag_1$	10	15	9	12
$ag_2$	14	8	11	13
$ag_3$	12	14	10	7
$ag_4$	9	12	15	11

Destination	Pen	Paper	Glue	Cutter
Desk A	11	16	9	8
Desk B	14	7	12	9

**Table 2.** Distances among locations

### 3 Background

#### 3.1 Dependence Networks and Coalition Formation

Our model for dependencies among agents in multiagent systems is based on dependence networks. According to Conte and Sichman [34], dependence networks can be used to represent the pattern of relationships that exist between agents, and more specifically, interdependencies among agents goals and actions. They can be used to study emerging social structures such as aggregates of heterogeneous agents. They are based on a social reasoning mechanism, on social dependence and on power [32]. Power, in this context, means the ability to fulfill a goal. Multi-agent dependence allows one to express a wide range of interdependent situations between agents.

A dependence network consists of a finite set or sets of actors and the relation or relations between them [33]. Actors can be people or organizations. They are linked together by goals, behaviors and exchanges such as hard currency or information. The structural similarity between dependence networks and directed graphs is such that a dependence network can be represented as a dependence graph. Informally, the nodes in the graph represent both the agents themselves, and the actions they have to perform to reach a goal. The directed edges in the graph are labelled with goals, and link agents with actions.

When agents cooperate to achieve some of their goals, they form groups or coalitions. Coalitions are topological aspects of a dependence network. They are indicative of some kind of organization, for example, the cooperation between agents in the dependence network. The coalition is supposed to ensure individual agents a sufficient payoff to motivate them to collaborate. In a coalition, agents coordinate their behaviors to reach their shared or reciprocal goals, for example in [27, 34]. All the agents in the coalition somehow benefit from the goals being reached. A coalition can achieve its purpose if its members are cooperative, i.e., if they adopt the goals of the coalition in addition to their own goals.

### 3.2 Multi-Context Systems

Multi-Context Systems (MCS) [16, 15, 6] has been the main effort to formalize context and contextual reasoning in Artificial Intelligence. We use here the definition of heterogeneous nonmonotonic MCS given in [6]. The main idea is to allow different logics to be used in different contexts, and to model information flow among contexts via bridge rules. According to [6], a MCS is a set of contexts, each composed of a knowledge base with an underlying logic, and a set of bridge rules. A logic  $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$  consists of the following components:

- $\mathbf{KB}_L$  is the set of well-formed knowledge bases of  $L$ . Each element of  $\mathbf{KB}_L$  is a set of formulae.
- $\mathbf{BS}_L$  is the set of possible belief sets, where the elements of a belief set is a set of formulae.
- $\mathbf{ACC}_L: \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$  is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.

As shown in [6], this definition captures the semantics of many different logics both monotonic, e.g. propositional logic, description logics and modal logics, and nonmonotonic, e.g. default Logic, circumscription, defeasible logic and logic programs under the answer set semantics.

A *bridge rule* refers in its body to other contexts and can thus add information to a context based on what is believed or disbelieved in other contexts. Bridge rules are added to those contexts to which they potentially add new information. Let  $L = (L_1, \dots, L_n)$  be a sequence of logics. An  $L_k$ -bridge rule  $r$  over  $L$ ,  $1 \leq k \leq n$ , is of the form

$$r = (k : s) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \text{not}(c_{j+1} : p_{j+1}), \dots, \text{not}(c_m : p_m). \quad (1)$$

where  $c_i$ ,  $1 \leq i \leq n$ , refers to a context,  $p_i$  is an element of some belief set of  $L_{c_i}$ , and  $k$  refers to the context receiving information  $s$ . We denote by  $h_b(r)$  the belief formula  $s$  in the head of  $r$ .

A MCS  $M = (C_1, \dots, C_n)$  is a set of contexts  $C_i = (L_i, kb_i, br_i)$ ,  $1 \leq i \leq n$ , where  $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$  is a logic,  $kb_i \in \mathbf{KB}_i$  a knowledge base, and  $br_i$  a set of  $L_i$ -bridge rules over  $(L_1, \dots, L_n)$ . For each  $H \subseteq \{h_b(r) | r \in br_i\}$  it holds that  $kb_i \cup H \in \mathbf{KB}_{L_i}$ , meaning that bridge rule heads are compatible with knowledge bases.

A belief state of a MCS is the set of the belief sets of its contexts. Formally, a *belief state* of  $M = (C_1, \dots, C_n)$  is a sequence  $S = (S_1, \dots, S_n)$  such that  $S_i \in \mathbf{BS}_i$ . Intuitively,  $S$  is derived from the knowledge of each context and the information conveyed through applicable bridge rules. A bridge rule of form (1) is applicable in a belief state  $S$  iff for  $1 \leq i \leq j$ :  $p_i \in S_{c_i}$  and for  $j < l \leq m$ :  $p_l \notin S_{c_l}$ . Equilibrium semantics selects certain belief states of a MCS  $M = (C_1, \dots, C_n)$  as acceptable. Intuitively, an equilibrium is a belief state  $S = (S_1, \dots, S_n)$  where each context  $C_i$  respects all bridge rules applicable in  $S$  and accepts  $S_i$ . Formally,  $S = (S_1, \dots, S_n)$  is an equilibrium of  $M$ , iff for  $1 \leq i \leq n$ ,

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) | r \in br_i \text{ applicable in } S\}).$$

[6] presents also an analysis on computational complexity, focusing on MCS with logics that have *poly-size kernels* such as propositional logic, propositional Defeasible Logic, Autoepistemic Logic and Nonmonotonic Logic Programs. According to this analysis, for a MCS  $M$ , deciding whether a literal  $p$  is in a belief set  $S_i$  for some (or each) equilibrium of  $M$  is in  $\Sigma_{k+1}^p$  (resp.  $\Pi_{k+1}^p = co - \Sigma_{k+1}^p$ ).

## 4 Computing and evaluating coalitions

One question that arises in scenarios such as the one that we present in Section 2 is how to compute the alternative coalitions that may be formed to achieve a set of given goals. Here we present a solution based on the use of heterogeneous nonmonotonic MCS [6], described in Section 3. The main reasons for choosing the MCS model are: (a) it enables representing systems consisting of agents with different knowledge representation models; (b) it can represent different kinds of relationships among agents such as goal-based dependencies, constraints and conflicting goals; and (c) it provides both centralized and distributed reasoning algorithms, which can be used for computing goal-based coalitions. Our solution consists, roughly, of representing agent dependencies and inter-agent constraints using *bridge rules* and computing the potential coalitions using algorithms for MCS equilibria.

### 4.1 Modeling dependencies

We model each agent in a multiagent system as a context in a MCS. The knowledge base of the context describes the goals of the agent and the actions that it

can perform. Goals and actions are represented as literals of the form  $g_k, a_j$ , respectively. Bridge rules represent the dependencies of the agent on other agents to achieve its goals. According to the definition given by [34], a dependence relation

$$dp : \text{basic\_dep}(ag_i, ag_j, g_k, p_l, a_m)$$

denotes that agent  $ag_i$  depends on agent  $ag_j$  to achieve goal  $g_k$ , because  $ag_j$  may perform action  $a_m$  needed in the plan  $p_l$ , which achieves the goal. For a goal  $g_k$  of agent  $ag_i$ , which is achieved through plan  $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$ , where  $ag_j : a_j$  represents action  $a_j$  performed by agent  $ag_j$ , the following dependence relations hold:

$$dp_j : \text{basic\_dep}(ag_i, ag_j, g_k, p_l, a_j), j = \{1, \dots, n\}$$

We denote this set of dependencies as  $DP(ag_i, g_k, p_l)$ . One way to represent dependencies is by using rules of the form:  $Head \leftarrow Body$ , where the *Head* denotes the goal of agent  $ag_i$  that is to be achieved ( $g_k$ ), and the *Body* describes the actions of plan  $p_l$  that will lead to the achievement of the goal. Based on this intuition, we define bridge rules representing dependence relations among agents as follows:

**Definition 1.** For an agent  $ag_i$  with goal  $g_k$  achieved through plan  $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$ , the set of dependencies  $DP(ag_i, g_k, p_l)$  is represented by a bridge rule of the form:

$$(c_i : g_k) \leftarrow (c_1 : a_1), (c_2 : a_2), \dots, (c_n : a_n)$$

where  $c_j, j = 1, \dots, i, \dots, n$  is the context representing agent  $ag_j$ .

Based on the above representation of agents as contexts, and goal-based dependencies among agents as bridge rules, we represent multiagent systems as MCS as follows.

**Definition 2.** A MCS  $M(A)$  corresponding to a multiagent system  $A$  is a set of contexts  $c_i = \{L_i, kb_i, br_i\}$ , where  $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$  is the logic of agent  $ag_i \in A$ ,  $kb_i \in \mathbf{KB}_i$  is a knowledge base that describes the actions that  $ag_i$  can perform and its goals, and  $br_i$  is a set of bridge rules, a subset of which represents the dependencies  $DP(ag_i, g_k, p_l)$  of  $ag_i$  on other agents in  $A$  for all goals  $g_k$  of  $ag_i$  and all plans  $p_l$ , with which these goals can be achieved.

*Example 1.* In our example, we assume that all four robots use propositional logic as their knowledge representation model. We model the four robots,  $ag_1$ - $ag_4$ , as contexts  $c_1$ - $c_4$ , respectively, with the following knowledge bases:

$$\begin{aligned} kb_1 &= \{a_{2s}, a_{1d}, a_{3d}, a_{1c} \vee a_{3c}\} \\ kb_2 &= \{a_{1s}, a_{4d}, a_{2c}\} \\ kb_3 &= \{a_{4s}, a_{2d}, a_{3c} \vee a_{4c}\} \\ kb_4 &= \{a_{3s}, a_{1c} \vee a_{4c}\} \end{aligned}$$

where  $a_{ij}$  represents the actions that a robot can perform.  $i$  stands for the object to be delivered: 1 stands for the pen, 2 for the paper, 3 for the glue and 4 for the cutter.  $j$  stands for the kind of action that the agent can perform:  $c$  stands for carrying the object,  $s$  stands for providing information about the current location (source) of the object, while  $d$  stands for providing information about the destination of the object. For example,  $ag_1$  can provide information about the source of the paper ( $a_{2s}$ ) and the destinations of the pen ( $a_{1d}$ ) and the glue ( $a_{3d}$ ), and can carry the pen and the glue ( $a_{1c} \vee a_{3c}$ ).

We represent the four tasks that the robots are requested to perform,  $t_i$ , as goals,  $g_i$ . For example  $g_1$  represents the task of delivering the pen to desk A ( $t_1$ ). We also assume that a robot  $ag_j$  can fulfil goal  $g_i$ , i.e. deliver object  $i$  to its destination, if it can perform action  $a_{ic}$ , i.e. carry object  $i$ . For example,  $g_1$  can be fulfilled by robots  $ag_1$  and  $ag_4$ , because these robots can carry the pen ( $a_{1c}$ ).

Given the knowledge and capabilities of robots, as described in Table 1, the robots can fulfil goals  $g_1 - g_4$  as follows. For  $g_1$ , there are two alternative plans:

$$\begin{aligned} p_{11} &= (ag_2 : a_{1s}, ag_1 : a_{1c}) \\ p_{12} &= (ag_2 : a_{1s}, ag_1 : a_{1d}, ag_4 : a_{1c}) \end{aligned}$$

According to  $p_{11}$ , robot  $ag_2$  must provide information about the source of the pen ( $ag_2 : a_{1s}$ ) and  $ag_1$  must carry the pen to its destination ( $ag_1 : a_{1c}$ ). According to  $p_{12}$ , robot  $ag_2$  must provide information about the source of the pen ( $ag_2 : a_{1s}$ ),  $ag_1$  must provide information about its destination ( $ag_1 : a_{1d}$ ), and  $ag_4$  must carry the pen to its destination ( $ag_4 : a_{1c}$ ).

For  $g_2$  there is only one plan,  $p_{21}$ ; for  $g_3$  there are two alternative plans:  $p_{31}$  and  $p_{32}$ ; and for  $g_4$  there are two plans as well:  $p_{41}$  and  $p_{42}$ :

$$\begin{aligned} p_{21} &= (ag_1 : a_{2s}, ag_3 : a_{2d}, ag_2 : a_{2c}) \\ p_{31} &= (ag_4 : a_{3s}, ag_1 : a_{3c}) \\ p_{32} &= (ag_4 : a_{3s}, ag_1 : a_{3d}, ag_3 : a_{3c}) \\ p_{41} &= (ag_2 : a_{4d}, ag_3 : a_{4c}) \\ p_{42} &= (ag_3 : a_{4s}, ag_2 : a_{4d}, ag_4 : a_{4c}) \end{aligned}$$

Each plan implies dependencies among robots. For example, from  $p_{11}$  the following dependency is derived:  $dp_1 : basic\_dep(ag_1, ag_2, g_1, p_{11}, a_{1s})$ , namely  $ag_1$  depends on  $ag_2$  to achieve goal  $g_1$ , because  $ag_2$  can provide information about the source of the pen ( $a_{1s}$ ). Figure 2 represents the dependencies derived from all plans, abstracting from plans, similarly to [32]. The figure should be read as follows: The pair of arrows going from node  $ag_1$  to the rectangle box labeled  $a_{1s}$  and then to node  $ag_2$  indicates that agent  $ag_1$  depends on agent  $ag_2$  to achieve goal  $g_1$ , because the latter can perform action  $a_{1s}$ .

Bridge rules  $r_1$ - $r_7$  represent the same dependencies. Each rule represents the dependencies derived by a different plan. For example  $r_1$  corresponds to plan



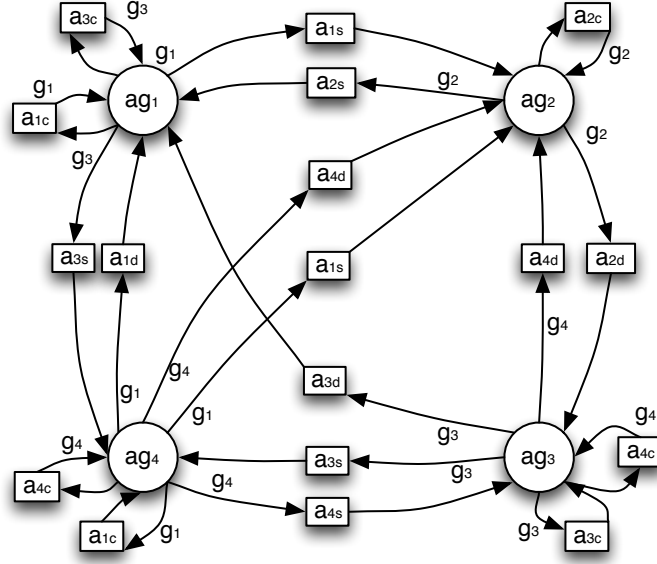


Fig. 2. Dependencies among the four robot agents.

$p_{11}$  and represents dependency  $dp_1$ .

$$\begin{aligned}
r_1 &= (ag_1 : g_1) \leftarrow (ag_1 : a_{1c}), (ag_2 : a_{1s}) \\
r_2 &= (ag_4 : g_1) \leftarrow (ag_4 : a_{1c}), (ag_2 : a_{1s}), (ag_1 : a_{1d}) \\
r_3 &= (ag_2 : g_2) \leftarrow (ag_2 : a_{2c}), (ag_1 : a_{2s}), (ag_3 : a_{2d}) \\
r_4 &= (ag_1 : g_3) \leftarrow (ag_1 : a_{3c}), (ag_4 : a_{3s}) \\
r_5 &= (ag_3 : g_3) \leftarrow (ag_3 : a_{3c}), (ag_4 : a_{3s}), (ag_1 : a_{3d}) \\
r_6 &= (ag_3 : g_4) \leftarrow (ag_3 : a_{4c}), (ag_2 : a_{4d}) \\
r_7 &= (ag_4 : g_4) \leftarrow (ag_4 : a_{4c}), (ag_3 : a_{4s}), (ag_2 : a_{4d})
\end{aligned}$$

One system constraint is that two robots cannot carry the same object at the same time. This can be described with bridge rules of the form:

$$\neg ag_l : a_{ic} \leftarrow ag_k : a_{ic}$$

where  $i, k, l = \{1 \dots 4\}$  and  $k \neq l$ . For example, the following rules describe that  $ag_1$  will not carry the pen if one of the other three robots is already carrying it.

$$\begin{aligned}
\neg(ag_1 : a_{1c}) &\leftarrow (ag_2 : a_{1c}) \\
\neg(ag_1 : a_{1c}) &\leftarrow (ag_3 : a_{1c}) \\
\neg(ag_1 : a_{1c}) &\leftarrow (ag_4 : a_{1c})
\end{aligned}$$

Note that using MCS enables us to represent agents that are heterogeneous with respect to the knowledge representation model that they use. In our running

example, we assumed (for reasons of simplicity) that the four agents use propositional logic. However, we can also represent any agent using a logic that can be captured by Definition 2. Note also that we use a rather simplistic representation for plans, because our goal is not to represent and reason with plans; we are only interested in the dependencies derived from plans.

## 4.2 Computing coalitions

An equilibrium in MCS represents an acceptable belief state of the system. Each belief set in this state is derived from the knowledge base of the corresponding context and is compatible with the applicable bridge rules. For a MCS  $M(A)$  that corresponds to a multiagent system  $A$ , an equilibrium  $S = \{S_1, \dots, S_n\}$  represents a coalition in which agents of  $A$  can achieve their goals. Specifically, each belief set  $S_i$  in the equilibrium contains the actions that agent  $ag_i$  can perform and the goals that it will achieve in this coalition. If there is more than one ways with which the goals can be achieved, the MCS will have more than one equilibria, each one representing a different coalition. If a certain goal does not appear in any of the equilibria, this means that there is no coalition with which the goal can be achieved.

In order to compute the potential coalitions in a multiagent system  $A$ , one then has to formulate the MCS  $M(A)$  that corresponds to  $A$ , and compute the equilibria  $S$  of  $M(A)$ . The computation of equilibria can either be done by a central entity that monitors the bridge rules of all agents [6]; or in a distributed fashion using the distributed algorithm proposed in [11].

*Example 2.* In our example, the MCS that corresponds to the system of the four robots,  $M(A)$ , has two equilibria:  $S_0$  and  $S_1$ :

$$\begin{aligned} S_0 = & \{ \{a_{2s}, a_{1d}, a_{3d}, a_{3c}, g_3\}, \{a_{1s}, a_{4d}, a_{2c}, g_2\}, \\ & \{a_{4s}, a_{2d}, a_{4c}, g_4\}, \{a_{3s}, a_{1c}, g_1\} \} \\ S_1 = & \{ \{a_{2s}, a_{1d}, a_{3d}, a_{1c}, g_1\}, \{a_{1s}, a_{4d}, a_{2c}, g_2\}, \\ & \{a_{4s}, a_{2d}, a_{3c}, g_3\}, \{a_{3s}, a_{4c}, g_4\} \} \end{aligned}$$

$S_0$  represents coalition  $C_0$ , according to which  $ag_1$  delivers the glue to desk B ( $g_3$ ),  $ag_2$  delivers the paper to desk A ( $g_2$ ),  $ag_3$  delivers the cutter to desk B ( $g_4$ ) and  $ag_4$  delivers the pen to desk A ( $g_1$ ).  $S_1$  represents coalition  $C_1$ , according to which  $ag_1$  delivers the pen to desk A ( $g_1$ ),  $ag_2$  delivers the paper to desk A ( $g_2$ ),  $ag_3$  delivers the glue to desk B ( $g_3$ ) and  $ag_4$  delivers the cutter to desk B ( $g_4$ ). Using the previous abstraction of plans, the two coalitions are graphically represented in Figure 3.

In order to achieve their goals, the robots then have to carry out the actions in the plans that are associated to these goals. For example, for coalition  $C_0$  the associated plans are:  $p_{12}$  (for goal  $g_1$ ),  $p_{21}$  (for  $g_2$ ),  $p_{31}$  (for  $g_3$ ) and  $p_{41}$  (for  $g_4$ ), while the plans associated to  $C_1$  are  $p_{11}$ ,  $p_{21}$ ,  $p_{32}$  and  $p_{42}$ .

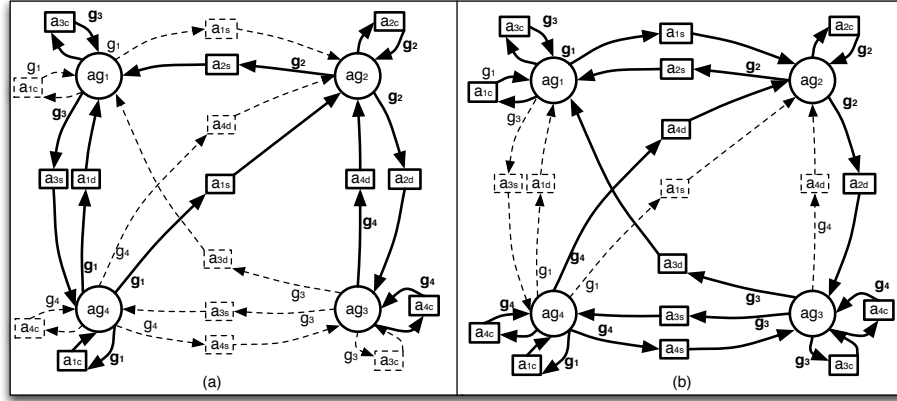


Fig. 3. Coalitions  $C_0$  (a), and  $C_1$  (b) in bold; remaining dependencies in dotted lines.

### 4.3 Selecting the best coalition

Selecting the best among the coalitions in which agents can achieve their goals requires evaluating and comparing them. Efficiency and stability metrics are commonly used to evaluate coalitions (e.g., [24, 29, 28, 19]). The former giving an assurance on the economical gain reached by being in the coalition, the later giving a certainty that the coalition is viable on the long term.

Generally speaking, efficiency in a coalition is a relation between what agents can achieve as part of the organization compared to what they can do alone or in different coalitions. Furthermore, a coalition is economically efficient iff i) no one can be made better off without making someone else worse off, ii) no additional output can be obtained without increasing the amount of inputs, iii) production proceeds at the lowest possible per-unit cost [24].

In our example, we can associate efficiency to the distances that the four robots must cover to perform the required actions. From Table 2 we can compute the distance for each robot to do each task, and, by adding them up, the cost of executing tasks in a given coalition. For instance, the cost of  $C_0$  is  $Cost(C_0) = 81$  whereas the cost of  $C_1$  is  $Cost(C_1) = 87$ . If we compare  $C_0$  and  $C_1$ ,  $C_0$  is economically efficient as at least one agent is better off without making anyone worse off, all else being equal;  $C_0$  is also more cost efficient than  $C_1$ .

Stability of coalitions is related to the potential gain in staying in the coalition or quitting the coalition for more profit (i.e., free riding). Hence, several elements come to play for the evaluation of a coalition's stability such as the characteristic function [23], Shapley value [29], nucleolus [28], Satisfactory Nucleolus [19] and others.

Depending on the application domain, other functional and non-functional requirements, e.g., security, user-friendliness or conviviality, may also play an important role in the choice of a coalition. In [8], we compared coalitions in terms of conviviality, which we measured by the number of reciprocity based

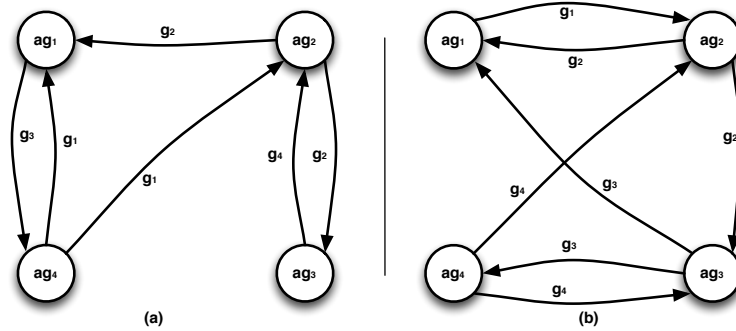
coalitions that can be formed within an overall coalition. Given the dependence network ( $DN$ ) that corresponds to a given coalition, the conviviality of the coalition  $Conv(DN)$  was measured as follows:

$$Conv(DN) = \frac{\sum coal(a, b)}{\Omega}, \quad (2)$$

$$\Omega = |A|(|A| - 1) \times \Theta, \quad (3)$$

$$\Theta = \sum_{L=2}^{L=|A|} P(|A| - 2, L - 2) \times |G|^L, \quad (4)$$

where  $|A|$  is the number of agents in the system,  $|G|$  is the number of goals,  $P$  is the usual permutation defined in combinatorics,  $coal(a, b)$  for any distinct pair of agents  $a, b \in A$  is the number of cycles that contain both  $a$  and  $b$  in  $DN$ ,  $L$  is the cycle length, and  $\Omega$  denotes the maximal number of pairs of agents in cycles.



**Fig. 4.** Goal dependencies in coalitions  $C_0$  (a), and  $C_1$  (b).

Abstracting from plans and actions, Figures 4.a and 4.b represent the dependence networks for coalitions  $C_0$  and  $C_1$  respectively. By applying formula 2 on the two dependence networks, we can compute the conviviality of the two coalitions:  $Conv(C_0) = 0.0000897$ ,  $Conv(C_1) = 0.000143$ .  $C_1$  is therefore preferred to  $C_0$  in terms of conviviality.

In cases of agents with conflicting goals, coalitions differ in the set of goals that they can fulfil and the selection of a coalition depends on the priorities among the conflicting goals. It is among our future plans to integrate in the proposed model a preference relation on the set of goals to represent such priorities and develop algorithms for preference-based coalition formation.

## 5 Related works

This is not the first work that brings together agents and context logics. [25] and [26] used Multi-Context Systems as a means of specifying and implementing agent architectures. Both studies propose *breaking* the logical description of an agent into a set of contexts, each of which represents a different component of the architecture, and the interactions between these components are specified by means of bridge rules between the contexts. [25] followed this approach to simplify the construction of a BDI agent, while [26] extended it to handle more efficiently implementation issues such as grouping together contexts in modules, and enabling inter-context synchronization. The main difference of our approach is that its focus is not on the internal representation of agents, but rather on their interactions with other agents and the coalitions that they can form.

Our previous work on evaluating information exchange in distributed information systems was based on modeling MCS as dependence networks where bridge rules are represented as dependencies [9]. Here we do the opposite: we use bridge rules to represent dependencies among agents, and model agents as contexts in MCS.

Several works from different research areas have focused on the problem of coalition formation including variants of the contract net protocol [14, 20], according to which agents break down composite tasks into simpler subtasks and subcontract subtasks to other agents via a bidding mechanism; formal approaches from multiagent systems, e.g. [18, 30]; and solutions from the field of robotics based on schema theory, e.g. [35, 36] or synergy [22]. The distinct characteristics of our approach are: (a) it allows agents to use different knowledge representation models; (b) based on a non-monotonic reasoning model, it enables representing and reasoning with agents with conflicting goals; and (c) it provides both centralized and distributed algorithms for computing coalitions, and can hence be applied in settings with different requirements for information hiding and sharing.

## 6 Summary and Future Work

In multiagent systems agents often depend on each other and need to cooperate in order to achieve their goals. In this paper we deal with the problem of computing the alternative coalitions in which the agents may fulfil their goals. Specifically, we propose a MCS-based representation of multiagent systems, in which agents are modeled as contexts, and dependence relations among agents as bridge rules. Based on this representation, we then compute the equilibria of the MCS, which represent the coalitions in which the agents may fulfil their goals. Finally, given a set of functional and non-functional requirements such as efficiency, stability and conviviality, we select the best coalitions. We demonstrate the proposed approach using an example from robotics, in which four different robots need to cooperate in order to perform a given set of tasks. For simplicity we assumed that all four robots use propositional logic. However, being based on

MCS, the proposed solution may also handle agents using different knowledge representation models.

In further research, we contemplate the need to integrate preferences on agents and goals into our model. Building on previous work on preference-based inconsistency resolution in MCS [2, 12, 13], we will develop algorithms for preference-based coalition formation in the presence of conflicting goals. We also plan to extend our approach with elements of dynamic MCS [10], i.e. schematic bridge rules that are instantiated at run time with concrete contexts. This will enable us handling changes such as the failure of an agent, the arrival of a new agent or any change in the operating environment. We will also look into applying and testing our methods in different kinds of agent-based systems characterized by heterogeneity of the participating agents, openness and dynamicity, such as ubiquitous robots and Ambient Intelligence systems. To achieve this we will use existing MCS implementations, such as DMCS<sup>1</sup>, a distributed solver for MCS, and MCS-IE<sup>2</sup>, a tool for explaining inconsistencies in MCS.

## References

1. Antoniou, G., Papatheodorou, C., Bikakis, A.: Reasoning about Context in Ambient Intelligence Environments: A Report from the Field. In: Principles of Knowledge Representation and Reasoning: Proceedings of the 12th Intl. Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010. pp. 557–559. AAAI Press (2010)
2. Bikakis, A., Antoniou, G., Hassapis, P.: Strategies for contextual reasoning with conflicts in Ambient Intelligence. *Knowledge and Information Systems* 27(1), 45–84 (2011)
3. Boella, G., Sauro, L., van der Torre, L.: Algorithms for finding coalitions exploiting a new reciprocity condition. *Logic Journal of the IGPL* 17(3), 273–297 (2009)
4. Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics* 1, 153–184 (2003)
5. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: International Semantic Web Conference. pp. 164–179 (2003)
6. Brewka, G., Eiter, T.: Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada. pp. 385–390 (2007)
7. Caire, P., Villata, S., Boella, G., van der Torre, L.: Conviviality masks in multiagent systems. In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 3. pp. 1265–1268 (2008)
8. Caire, P., Alcade, B., van der Torre, L., Sombattheera, C.: Conviviality measures. In: 10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011 (2011)
9. Caire, P., Bikakis, A., Traon, Y.L.: Information Dependencies in MCS: Conviviality-Based Model and Metrics. In: Principles and Practice of Multi-Agent Systems, PRIMA. pp. 405–412 (2013)

---

<sup>1</sup> <http://www.kr.tuwien.ac.at/research/systems/dmcs/>

<sup>2</sup> <http://www.kr.tuwien.ac.at/research/systems/mcsie/>

10. Dao-Tran, M., Eiter, T., Fink, M., Krennwallner, T.: Dynamic distributed non-monotonic multi-context systems. *Nonmonotonic Reasoning, Essays Celebrating its 30th Anniversary, Studies in Logic* 31
11. Dao-Tran, M., Eiter, T., Fink, M., Krennwallner, T.: Distributed nonmonotonic multi-context systems. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010* (2010)
12. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding Explanations of Inconsistency in Multi-Context Systems. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press (2010)
13. Eiter, T., Fink, M., Weinzierl, A.: Preference-Based Inconsistency Assessment in Multi-Context Systems. In: *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010*. Proceedings. Lecture Notes in Computer Science, vol. 6341, pp. 143–155. Springer (2010)
14. Gerkey, B.P., Mataric, M.J.: Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation, Special Issue on Multi-Robot Systems* 18(5), 758–768 (Oct 2002), <http://robotics.usc.edu/publications/10/>, (Also Technical Report IRIS-01-399)
15. Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=locality+compatibility. *Artificial Intelligence* 127(2), 221–259 (2001)
16. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence* 65(1) (1994)
17. Grossi, D., Turrini, P.: Dependence theory via game theory. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S. (eds.) *AAMAS*. pp. 1147–1154. IFAAMAS (2010)
18. Klusch, M., Gerber, A.: Dynamic coalition formation among rational agents. *IEEE Intelligent Systems* 17(3), 42–47 (2002)
19. Kronbak, L.G., Lindroos, M.: Sharing rules and stability in coalition games with externalities. *Marine Resource Economics* 22, 137–154 (2007)
20. Lemaire, T., Alami, R., Lacroix, S.: A distributed tasks allocation scheme in multi-uav context. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*. pp. 3622–3627 (2004)
21. Lenat, D.B., Guha, R.V.: *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)
22. Liemhetcharat, S., Veloso, M.M.: Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artif. Intell.* 208, 41–65 (2014)
23. Mesterton-Gibbons, M.: *An Introduction to Game-theoretic Modelling*. Addison-Wesley, Redwood, CA, USA (1992)
24. O’Sullivan, A., Sheffrin, S.M.: *Economics: Principles in Action*. Pearson Prentice Hall (2006)
25. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3), 261–292 (1998)
26. Sabater, J., Sierra, C., Parsons, S., Jennings, N.R.: Engineering Executable Agents using Multi-context Systems. *Journal of Logic and Computation* 12(3), 413–442 (2002)
27. Sauro, L.: *Formalizing Admissibility Criteria in Coalition Formation among Goal Directed Agents*. Ph.D. thesis, University of Turin, Italy (2006)

28. Schmeidler, D.: The nucleolus of a characteristic functional game. *SIAM journal of Applied Mathematics* 17, 1163–1170 (1969)
29. Shapley, L.S.: A value for n-person games. *Annals of Mathematical Studies* 28, 307–317 (1953)
30. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. *Artificial Intelligence* 101(1-2), 165–200 (1998)
31. Sichman, J.S.: Depint: Dependence-based coalition formation in an open multi-agent scenario. *J. Artificial Societies and Social Simulation* 1(2) (1998)
32. Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: *Proc. of The First Int. Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*. pp. 483–490. ACM (2002)
33. Sichman, J.S., Conte, R., Castelfranchi, C., Demazeau, Y.: A social reasoning mechanism based on dependence networks. In: *Proceedings of the Eleventh European Conference on Artificial Intelligence, Amsterdam, The Netherlands, August 8-12, 1994*. pp. 188–192 (1994)
34. Sichman, J.S., Demazeau, Y.: On social reasoning in multi-agent systems. *Revista Iberoamericana de Inteligencia Artificial* 13, 68–84 (2001)
35. Tang, F., Parker, L.E.: Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*. pp. 1501–1508 (2005)
36. Zhang, Y., Parker, L.E.: Iq-asymtre: Forming executable coalitions for tightly coupled multirobot tasks. *IEEE Transactions on Robotics* 29(2), 400–416 (2013)