Tools for Conviviality in Multi-Context Systems

Antonis Bikakis
Department of Information Studies, University College London, UK
a.bikakis@ucl.ac.uk

Patrice Caire and Yves Le Traon
University of Luxembourg, Interdisciplinary Center for Security, Reliability and Trust, Luxembourg
{patrice.caire,yves.letraon}@uni.lu

Abstract

A common feature of many distributed systems, including web social networks, peer-to-peer systems and Ambient Intelligence systems, is cooperation in terms of information exchange among heterogeneous entities. In order to facilitate the exchange of information, we first need ways to evaluate it. The concept of conviviality was recently proposed for modeling and measuring cooperation among agents in multiagent systems. In this paper, we introduce conviviality as a property of Multi-Context Systems (MCS). We first present how to use conviviality to model and evaluate interactions among different contexts, which represent heterogeneous entities in a distributed system. Then, as one cause of logical conflicts in MCS is due to the exchange of information between mutually inconsistent contexts, we show how inconsistency can be resolved using the conviviality property. We illustrate our work with an example from web social networks.

1 Introduction

Multi-Context Systems (*MCS*) [19, 18, 9] are logical formalizations of distributed context theories connected through a set of bridge rules, which enable information flow between contexts. A *context* can be thought of as a logical theory - a set of axioms and inference rules - that models local knowledge. Intuitively, MCS can be used as a representation model for any information system that involves distributed, heterogeneous knowledge agents such as peer-to-peer systems, distributed ontologies (e.g., Linked Open Data) or Ambient Intelligence systems. In fact, several applications have already been developed on top of MCS or other logic-based context formalizations including (a) the CYC common sense knowledge base [23]; (b) contextualized ontology languages, such as Distributed Description Logics [5] and C-OWL [6]; (c) context-based agent architectures [24, 25]; and (d) distributed reasoning algorithms for Mobile Social Networks [1] and Ambient Intelligence systems [2].

The individual entities that such systems consist of cooperate by sharing information. By reasoning with the information they import they are able to derive new knowledge. These features are enabled by the notions of *contexts*, *bridge rules* and *contextual reasoning* used in MCS. But, how can we then evaluate the ways in which a system enables this cooperation? How can we characterise a MCS based on the opportunities for information exchange that it provides to its contexts? To answer such questions, we build on previous work on modeling *conviviality* in a version of MCS called Contextual Defeasible Logic [12]. Here we extend these results for the general MCS model, and introduce measures for information dependencies based again on the notion of conviviality.

Defined by Illich as "individual freedom realized in personal interdependence" [21], conviviality was introduced as a social science concept for multiagent systems to highlight soft qualitative requirements like user friendliness of systems. Multiagent systems technology can be used to realize tools for conviviality when "freedom" is interpreted as choice [10]. Tools for conviviality are concerned in particular with dynamic aspects of conviviality, such as the emergence of conviviality from the sharing of properties or behaviors whereby each member's perception is that their personal needs are taken care of.

Conviviality is measured by counting the possible ways to cooperate, indicating degree of choice or freedom to engage in coalitions [11]. The authors' coalitional theory is based on dependence networks [13, 28], labeled directed graphs where nodes represent agents, (thus the graph represents a social network), and each labeled edge represents that the former agent depends on the latter one to achieve some goal (represented by the label).

The focus on dependence networks and specifically on their cycles is a reasonable way of formalizing conviviality as something related to the freedom of choice of individuals plus the subsidiary relations – interdependence for task achievement – among fellow members of a social system. In distributed information systems, individual freedom is linked to an agent's choice to keep personal knowledge and beliefs at the local level, while interdependence is understood as reciprocity, i.e. cooperation. Participating human and artificial entities depend on each other to increase their local knowledge.

In this paper, we draw a parallel between, on the one hand an agent and a

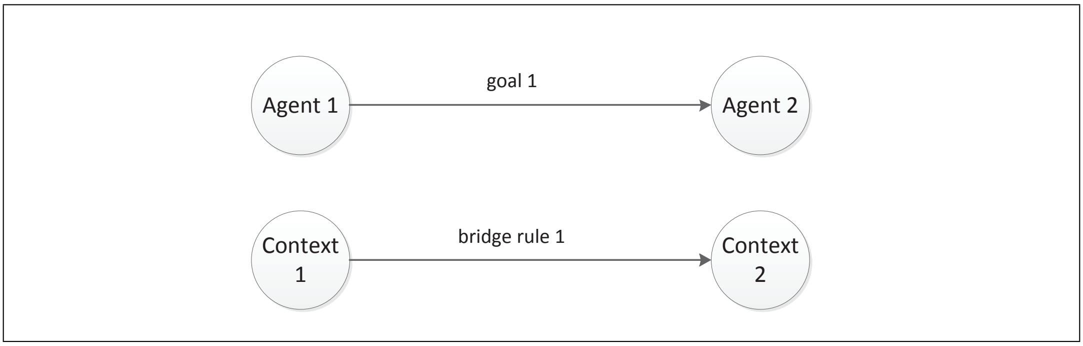

Figure 1: The dependence network parallelism of contexts as agents, and bridge rules as goals. A labeled arrow, from *Agent*1 to *Agent*2 means that the former depends on the latter to achieve its *goal*1.

context, and on the other hand a goal and a bridge rule. More specifically, we use a context to encode an agent's knowledge in some logic language, and a bridge rule to describe how an agent achieves its goal, namely to acquire and combine knowledge from other agents in order to deduce new knowledge, as illustrated in Figure 1. Therefore, the dependency from an agent A_1 towards a distinct agent A_2 to fulfill its goal g_1 corresponds to the context C_1 depending on a distinct context C_2 to acquire knowledge through the exchange of information described in a rule r_1. Furthermore, evaluating this exchange would allow to reason about the system with respect to how it can be reconfigured to enable more cooperation among contexts and thereby more information sharing, opportunities to collaborate and possibility to choose among them. Particularly, considering the potential applications of MCS, and the tools for conviviality described above, we formulate our main research question as follows:

> *How to evaluate and improve the exchange of information in systems modeled as MCS using conviviality modeling and measures?*

Our main research question breaks into the following questions:

1. How to define and model conviviality for Multi-Context Systems?

2. How to measure the conviviality of Multi-Context Systems?

3. How to use conviviality as a property of Multi-Context Systems?

In this paper we address these questions by proposing the following:

1. A formal model for representing *information dependencies* in MCS based on dependence networks,

2. Conviviality measures for MCS and

3. A potential application of these tools for the problem of inconsistency resolution in MCS.

So far, most approaches for inconsistency resolution in MCS have been based on the *invalidation* or *unconditional application* of a subset of the bridge rules that cause inconsistency. They differ in the preference criterion that is applied for selecting among the candidate solutions. In this work, we propose to use the conviviality of the system as a preference criterion, based on the idea that removing (or applying unconditionally) a bridge rule affects the information dependency between the connected contexts, and, as a result, the conviviality of the system. We suggest that the optimal solution is the one that minimally reduces conviviality.

The paper is structured as follows: Section 2 describes our running example from the social web application domain. Section 3 presents formal definitions for MCS, as these were originally proposed in [9]. Section 4 introduces a model and measures for conviviality in MCS. Section 5 proposes a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. Section 6, presents and compares related works. The last section summarizes and provides insights on our future works.

2 Running Example

In order to demonstrate the exchange of information among heterogeneous agents, we use an example from the domain of social networks, namely a social web application, and highlight the requirements and challenges with respect to knowledge sharing and collective decision making.

A typical challenge for students is to find how to organize their references. They need to record their readings and have quick and easy access to research articles. Therefore articles need to be classified in a way that is tailored to their studies. Furthermore, if more students contribute to this classification, more articles will be available to the whole group for citations.

Jane, Bob and Charlie are members of uni.scholar.space. They use software agents ($A1$, $A2$ and $A3$ respectively) to connect to the network in order to share information and classify research articles that they find online. The three agents are heterogeneous with respect to their capabilities, the knowledge that they encode, and the *logic* with which they represent and reason with the available knowledge. $A1$ retrieves the keywords of articles and encodes this information as well as Jane's research knowledge in propositional logic. $A2$ uses propositional logic as well to

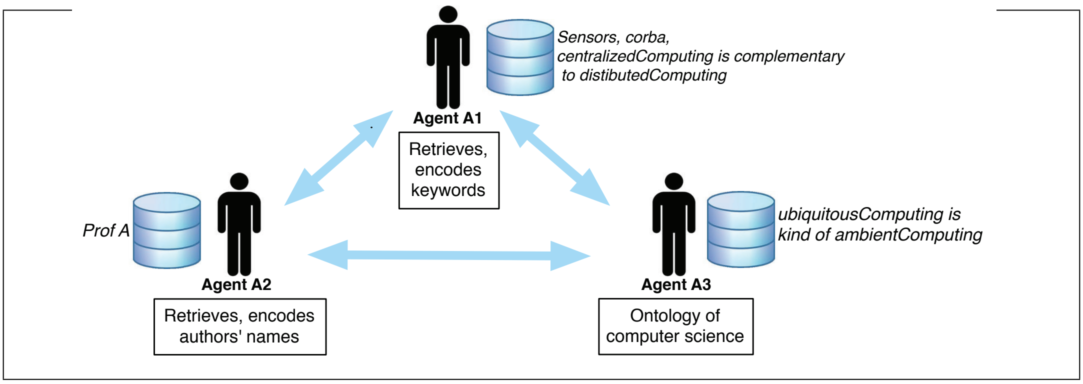

Figure 2: Students' social network for information exchange and collective classification of articles; ideal case where all agents cooperate with each other.

encode information about the authors of the articles and Bob's research knowledge. Finally, $A3$ contains an ontology about Computer Science written by Charlie in a basic description logic.

The system enables the heterogeneous agents to exchange their local knowledge, and take together decisions about the classification of an article by exploiting as much as possible from the available information. Figure 2 illustrates the ideal case where all agents are enabled to exchange information with each other and take a collective decision about the classification of an article.

To make the example more concrete we consider a specific article for which the three agents have retrieved the following metadata: the article has two keywords, $sensors$ and *corba*, and is written by $Prof.A$. Moreover, according to $A1$, centralized computing and distributed computing are two complementary concepts; and according to $A3$ ubiquitous computing is a form of ambient computing. In order to be able to exchange information, the three users have identified the following mappings between the concepts that they use: for Jane ($A1$) the term *middleware* used by Bob ($A2$) implies centralized computing, while the term *ambient computing* used by Charlie ($A3$) implies distributed computing. Bob knows that *corba* stands for Common Object Request Broker Architecture, and is a type of middleware. Finally, for Charlie, articles that are written by Prof. B and are about $sensors$ are relevant to ubiquitous computing.

In the following sections we show how information exchange between heterogeneous agents such as the ones in our running example is enabled by MCS; and how we can evaluate cooperation between agents in terms of opportunities for information exchange by using conviviality as a property of MCS.

3 Multi-Context Systems

We use here the definition of heterogeneous nonmonotonic MCS given in [7]. The idea behind heterogeneous MCSs is to allow different logics to be used in different contexts, and to model information flow among contexts via bridge rules. According to [7], a MCS M 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.
- $\mathbf{ACC}_L$: $\mathbf{KB}_L \rightarrow 2^{\mathrm{BS}_L}$ is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.

As shown in [7], this definition captures the semantics of many different logics both monotonic, e.g. propositional logic, description logics and modal logics, and non-monotonic, 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 $L = (L_1, \ldots, L_n)$ be a sequence of logics. An L_k-bridge rule r over L, $1 \leq k \leq n$, is of the form

$$\begin{aligned} r = (k : s) \leftarrow & (c_1 : p_1), \ldots, (c_j : p_j), \\ & \mathbf{not}(c_{j+1} : p_{j+1}), \ldots, \mathbf{not}(c_m : p_m). \end{aligned} \tag{1}$$

where c_i, $1 \leq c_i \leq n$, refers to a context in M, 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. By $br_M = \bigcup_{i=1}^{n} br_i$ we denote the set of bridge rules in M.

A *MCS* $M = (C_1, \ldots, C_n)$ 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 $(L_1, \ldots, L_n)$. 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.

Example 3.1. Agents $A1$, $A2$ and $A3$ of our running example can be modeled as contexts C_1, C_2 and C_3 respectively in a MCS $M = \{C_1, C_2, C_3\}$. The knowledge bases of the three contexts are:

$$\begin{aligned} kb_1 =& \{sensors, corba, centralizedComputing \leftrightarrow \neg distributedComputing\} \\ kb_2 =& \{profA\} \\ kb_3 =& \{ubiquitousComputing \subseteq ambientComputing\} \end{aligned}$$

The bridge rules that the three agents use to exchange information and collectively decide about the classification of the article are as follows:

$$\begin{aligned} r_1 =& (1 : centralizedComputing) \leftarrow (2 : middleware) \\ r_2 =& (1 : distributedComputing) \leftarrow (3 : ambientComputing) \\ r_3 =& (2 : middleware) \leftarrow (1 : corba) \\ r_4 =& (3 : ubiquitousComputing) \leftarrow (1 : sensors), (2 : profB) \end{aligned}$$

A belief state of a MCS is the set of the belief sets of its contexts. Formally, a *belief state* of $M = (C_1, \ldots, C_n)$ is a sequence $S = (S_1, \ldots, S_n)$ 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 as acceptable. Intuitively, for a MCS $M = (C_1, \ldots, C_n)$, an equilibrium is a belief state $S = (S_1, \ldots, S_n)$ where each context C_i respects all bridge rules that are applicable in S and accepts S_i. Formally, S is an equilibrium of M, iff for $1 \leq i \leq n$,

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

Example 3.2. In our running example, $S = (S_1, S_2, S_3)$ is the only equilibrium of the system:

$$S = (\{sensors, corba, centralizedComputing\}, \{profA, middleware\}, \emptyset).$$

S_3 is an empty set, since r_4, which is the only bridge rule in C_3, is not applicable in S, because $profB \notin S_2$.

4 The Conviviality Property in MCS

We recall from Section 1, that dependence networks have been proposed as a model for representing social dependencies among the agents of a multiagent system. They have also been used as the underlying model for formalizing and measuring conviviality in such systems. In this section, we describe how dependence networks can be used to model information dependencies among the contexts of a MCS and how conviviality measures can then be applied to MCS.

Our approach is based on the following ideas. First, cooperation in MCS can be understood as information sharing among its contexts. Second, this cooperation is enabled by the bridge rules of the system. Hence, finally, bridge rules actually represent information dependencies among contexts. On one hand, the more bridges between the contexts, the more possibilities for cooperation and information exchange. On the other hand, no bridge rules would mean that the different contexts represent autonomous systems, which do not share their local knowledge.

4.1 Model

Conviviality can be modeled by the reciprocity-based coalitions, or group of agents, that may be formed [11]. Some coalitions, however, provide more opportunities for their participants to cooperate than others, being thereby more convivial. Dependence networks are used to represent the interdependencies among the participants of the coalitions. Abstracting from tasks and plans that agents may have to achieve their goals, a dependence network for a multiagent system is defined [11] as follows:

Definition 4.1 (Dependence network)**.** *A dependence network (DN) is a tuple* $\langle A, G, dep, \geq \rangle$ *where:* A *is a set of agents,* G *is a set of goals,* $dep : A \times A \rightarrow 2^G$ *is a function that relates with each pair of agents, the sets of goals on which the first agent depends on the second, and* $\geq: A \rightarrow 2^G \times 2^G$ *is for each agent a total pre-order on sets of goals occurring in its dependencies:* $G_1 >_{(a)} G_2$*.*

To capture the notions of *context* and *bridge rule*, we build on Definition 4.1 and introduce a new definition, Definition 4.2, for a dependence network that corresponds to a MCS, as follows:

Definition 4.2 (Dependence network for MCS)**.** *A dependence network corresponding to a MCS* M*, denoted as* $DN(M)$*, is a tuple* $\langle C, br_M, dep, \geq \rangle$ *where:* C *is the set of contexts in* M*;* br_M *is the set of bridge rules in* M*;* $dep : C \times C \rightarrow 2^{br_M}$ *is a function that is constructed as follows: for each bridge rule* r *(in the form of (1)) in* br_M *add the following dependencies:* $dep(k, c_i) = \{r\}$ *where* k *is the context appearing in the head of* r *and* c_i *stands for each distinct context appearing in the body of* r*; and* $\geq: C \rightarrow 2^{br_M} \times 2^{br_M}$ *is for each context a total pre-order on sets of its bridge rules.*

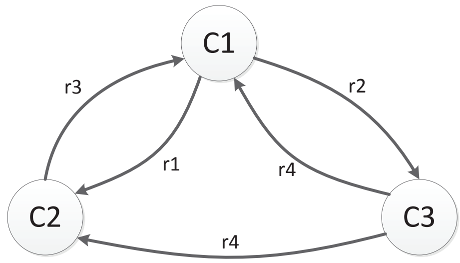

Figure 3: The dependence network $DN(M)$ of MCS M of the running example, in which a specific article is under examination.

In other words, a bridge rule r creates one dependency between context k, which appears in the head of r, and each of contexts c_i that appear in the body of r. The intuition behind this is that k depends on the information it receives from each c_i to achieve its goal, which is to apply r in order to infer s.

We should also note here that the total preorder that each context defines on the sets of bridge rules may reflect the local preferences of a context, e.g., in the way that these are defined and used in Contextual Defeasible Logic [2]. For sake of simplicity, we do not use this feature in the conviviality model that we describe in this paper. However, it is among our plans to integrate it in future extensions of this work. To graphically represent dependence networks, we use nodes for contexts and labeled arrows for dependencies among the contexts that the arrows connect. An arrow from context a to context b, labeled as r, means that a depends on b to apply rule r.

Example 4.3. In our running example, the dependence network that corresponds to MCS M is $DN(M) = \langle C, br_M, dep, \geq \rangle$ where:

- $C = \{C_1, C_2, C_3\}$ *is the set of contexts in* M
- $br_M = \{r_1, r_2, r_3, r_4\}$ *is the set of bridge rules in* M
- *The dependencies, as per Definition 4.2, are the following:* $dep(C_1, C_3) = \{r_2\}$, $dep(C_3, C_1) = \{r_4\}$, $dep(C_1, C_2) = \{r_1\}$, $dep(C_2, C_1) = \{r_3\}$, $dep(C_3, C_2) = \{r_4\}$;

The graphical representation of the dependence network is illustrated in Figure 3. The figure should be read as follows: each node corresponds to one of the contexts in M. Dependencies are derived from the four bridge rules of M. For example, there are two dependencies labeled by r_4: each of them connects C_3, which appears in the head of r_4, to each distinct context appearing in the body of r_4, namely C_1 and C_2 respectively. This actually means that to apply rule r_4 in order to prove that the paper under examination is about ubiquitous computing, C_3 depends on information about the keywords of the paper that it imports from C_1 and information about the authors of the paper that it imports from C_2.

To evaluate MCS in terms of the information exchange we introduce appropriate measures in the next section.

4.2 Measures

Conviviality measures were introduced to compare the conviviality of multiagent systems [11], for example before and after making a change such as adding a new norm or policy. Furthermore, to evaluate conviviality in a more precise way, the authors introduced formal conviviality measures for dependence networks using a coalitional game theoretic framework. Based on Illich's definition of conviviality as "individual freedom realized in personal interdependency", the notions of interdependency and choice, if freedom is interpreted as choice, are emphasized. Such measures provide insights into the type of attributes that may be measured in a convivial system and thus evaluate the quality of the system from this point of view. The conviviality measures we present here reflect the following hypotheses:

- H1 *The cycles identified in a dependence network are considered as coalitions, i.e., grouping of contexts. Such coalitions are used to evaluate conviviality in the network. Cycles are the smallest graph topology expressing interdependence, thereby conviviality, and are therefore considered atomic relations of interdependence. When referring to* cycles*, we implicitly signify* simple cycles*, i.e., where all nodes are distinct[14]; we also discard self-loops and logical loops. When referring to conviviality, we refer to potential interaction, not actual interaction.*

H1 is based on two intuitions: (a) bridge rules represent potential ways of information exchange (actual information exchange occurs only when such rules are applied); and (b) self-loops, which are created by bridge rules that contain elements of the same context in their heads and bodies, and logical loops (e.g. a loop created by two bridge rules of the form: $r_1 = (C_1 : a) \leftarrow (C_2 : b)$ and $r_2 = (C_2 : b) \leftarrow (C_1 : a)$) do not actually enable information exchange between contexts, and should not therefore be taken into account when measuring conviviality.

- H2 *Conviviality in a dependence network is evaluated in a bounded domain, i.e., over a $[min, max]$ interval. This allows to read the values obtained by any evaluation method.*

This allows the comparison of different systems in terms of conviviality.

H3 *There is more conviviality in larger coalitions than in smaller ones.*

The intuition for H3 is that a greater number of collaborating contexts in a MCS offers a greater source of knowledge. This means that a large coalition of contexts can reach more informative conclusions or take more informative decisions compared to smaller coalitions.

H4 *The more coalitions in the dependence network, the higher the conviviality measure (ceteris paribus).*

H4 reflects the fact that the number of opportunities for information exchange for a context increases with the number of coalitions that the context participates in, which, in turn, increases with the number of bridge rules defined in this context.

Some coalitions provide more opportunities for their participating contexts to cooperate than others, being thereby more convivial. For a cooperative system modeled as MCS, the top goal should be to maximize conviviality. It should, therefore, fulfil the following two requirements:

R1 *Maximize the size of the coalitions, i.e., maximize the number of contexts involved in the coalitions.*

R2 *Maximize the number of these coalitions.*

Based on requirements $R1$ and $R2$, we define the *conviviality of a MCS M* as:

$$\Theta = \sum_{L=2}^{L=|C|} P(|C|-2, L-2) \times d_M^L, \tag{2}$$

$$\Omega = |C|(|C|-1) \times \Theta, \tag{3}$$

$$\mathrm{Conv}(M) = \frac{\sum\limits_{c_i, c_j \in C, i \neq j} \mathrm{coal}(c_i, c_j)}{\Omega} \tag{4}$$

where $|C|$ is the number of contexts in M, L is the cycle length, P is the usual permutation defined in combinatorics, $\mathrm{coal}(c_i, c_j)$ for any distinct $c_i, c_j \in C$ is the number of cycles that contain the ordered pair (c_i, c_j) in $DN(M)$, such that the cycles do not represent logical loops, and Ω denotes the maximal number of pairs of contexts in cycles (which produces the normalization mentioned in Hypothesis H2).

d_M is the maximum number of dependencies that a context in M may have on other contexts of M:

$$d_M = \max_{k \in M} \sum_{i=1}^{|C|} dep(k, c_i) \tag{5}$$

Example 4.4. The dependence network of M, which is graphically represented in Figure 3 has three cycles: $\{(C_1, C_2, r_1), (C_2, C_1, r_3)\}$, $\{(C_1, C_3, r_2), (C_3, C_1, r_4)\}$ and $\{(C_1, C_3, r_2), (C_3, C_2, r_4), (C_2, C_1, r_3)\}$. The ordered pair (C_1, C_2) is only in the first cycle, therefore $coal(C_1, C_2) = 1$. In the same way we calculate $coal(C_2, C_1) = 2$, $coal(C_1, C_3) = 2$, $coal(C_3, C_1) = 1$, $coal(C_2, C_3) = 0$, $coal(C_3, C_2) = 1$. Following Equation 2 and assuming that $d_M = 1$, we calculate the conviviality of M as:

$$\text{Conv}(M) = 7/\Omega = 0.58, \text{ where } \Omega = 12.$$

We note that $Conv(M)$ is almost maximal as adding only one bridge rule, namely from C_2 to C_3, results in a fully connected graph, i.e., maximal conviviality.

Computational complexity: For our measures, the number of cycles going through every possible pair of contexts is needed. The computational complexity for counting cycles can be computed using first the measures based on graph properties, that is in $O(|C|+|br_M|)$. Then, for each pair and cycle, a check must be performed to evaluate if the pair is in the cycle. Therefore, the complexity is $O((|C||C-1|)(|C|+|br_M|))$.

In the next section we show how one can use conviviality measures for MCS to compare different states of a distributed information system and improve it in terms of cooperativeness.

5 Inconsistency Resolution

As we previously argued, conviviality is a property that characterizes the cooperativeness of a MCS, namely the alternative ways in which the agents can share information in order to derive new knowledge. By evaluating conviviality, we are able to propose different ways in which cooperation can be increased, e.g., by suggesting new connections between the agents - or in other words mappings between their contexts. Consider, for example, a system in which an agent does not import data from any other agent. Recommending other agents from which the first agent can potentially import information from, can increase the conviviality of the system, which will in turn lead not only to enriching the local knowledge of the agent, but also the knowledge of the whole system.

5.1 Problem Description

Another way of using conviviality as a property of MCS, which we describe in more detail in this section, is for the problem of inconsistency resolution. In a MCS, even if contexts are locally consistent, their bridge rules may render the whole system inconsistent. This is formally described in [9] as a *lack of an equilibrium.* All techniques that have been proposed so far for inconsistency resolution are based on the same intuition: a subset of the bridge rules that cause inconsistency must be invalidated and another subset must be unconditionally applied, so that the entire system becomes consistent again. For nonmonotonic MCS, this has been formally defined in [15] as diagnosis:

"Given a MCS M, a *diagnosis* of M is a pair (D_1, D_2), $D_1, D_2 \subseteq br_M$, s.t. $M[br_M \backslash D_1 \cup heads(D_2)] \not\models \bot$". $D^{\pm}(M)$ is the set of all such diagnoses, while with $M[R]$ we denote the MCS obtained from M by replacing its bridge rules br_M with R; therefore $M[br_M \backslash D_1 \cup heads(D_2)]$ is the MCS obtained from M by removing the rules in D_1 and adding the heads of the rules in D_2.

In other words, if we deactivate the rules in D_1 and apply the rules in D_2 in unconditional form, M will become consistent. In a MCS it is possible that there is more than one diagnosis that can restore consistency.

Example 5.1. In our running example, consider the case that $profB$ is also identified by C_2 as one of the authors of the paper under examination. In this case kb_2 would also contain $profB$: $kb_2 = \{profA, profB\}$.

This addition would result in an inconsistency in kb_1, caused by the activation of rules r_4 and r_2. Specifically, rule r_4 would become applicable, *ubiquitousCom−puting* and *ambientComputing* would become *true* in C_3, r_2 would then become applicable too, and *distributedComputing* would become *true* in C_1 causing an inconsistency with *centralizedComputing*, which has also been evaluated as *true*. To resolve this conflict, one of the four bridge rules r_1-r_4 must be invalidated. Using the definition of diagnosis that we presented above, this is formally described as:

$$D^{\pm}(M) = \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_3\}, \emptyset), (\{r_4\}, \emptyset)\}.$$

Various criteria have been proposed for selecting a diagnosis including: *i.*) the number of bridge rules contained in the diagnosis - specifically in [15] pointwise subset-minimal diagnoses are preferred, *ii.*) local preferences on diagnoses proposed in [16] and *iii.*) local preferences on contexts and provenance information used in Contextual Defeasible Logic [2].

5.2 Proposed Solution

We propose using the conviviality of the resulted system as a criterion for selecting a diagnosis. This actually means that for each diagnosis, we measure the conviviality of the system that is derived after applying the diagnosis, and select the diagnosis that minimally decreases conviviality. The intuition is that the system should remain as *cooperative* as possible, and this is achieved by maximizing the amount of agents involved in the derivation of a conclusion or a decision and the number of potential ways in which a conclusion may be drawn. In the extreme case of invalidating all bridge rules, there will be no inconsistencies; however the agents will not able to take collective decisions - they will decide based on their local knowledge only. Overall, we propose resolving inconsistencies, by also keeping as many bridge rules (hence possibilities for information exchange) as possible.

Diagnoses contain two types of changes applicable in the bridge rules: invalidation (removal) of a rule; and applying a rule unconditionally, which means removing the body of the rule. These changes affect the dependencies of the system as follows: When invalidating or adding unconditionally rule r (as defined in (1)) in a MCS M, all the dependencies labeled by r are removed from the dependence network of M.

Assuming that $D_i = (D_{i1}, D_{i2})$ is a diagnosis that we can apply in a MCS M, and $M(D_i)$ is the MCS obtained M after applying D_i, the optimal diagnosis is the one that maximizes the conviviality of $M(D_i)$:

$$D_{opt} = \{D_i : \mathrm{Conv}(M(D_i)) = max\}$$

Example 5.2. In the running example, there are four diagnoses that we can apply: D_1-D_4. Each of them requires invalidating one of rules r_1 to r_4, respectively. Figures 4-7 depict the four dependence networks $DN(M(D_i))$, which are derived after applying D_i. Dashed arrows represent the dependencies that are dropped in each $DN(M(D_i))$ compared to $DN(M)$.

Following Equation 2 and the four dependence networks (Figures 4-7) the conviviality of each DN is:

$$\begin{aligned}&Conv(M(D_1)) = 5/\Omega = 0.42 \text{ and}\\ &Conv(M(D_j)) = 2/\Omega = 0.17 \text{ with } j = 2, 3, 4 \text{ and } \Omega = 12\end{aligned}$$

By applying D_1 (Figure 4), only one cycle $\{(C_1, C_2, r_1), (C_2, C_1, r_3)\}$ is removed from the initial dependence network $DN(M)$. However, by applying any of diagnoses D_2-D_4 (Figures 5-7), two cycles are removed from $DN(M)$. Therefore the optimal

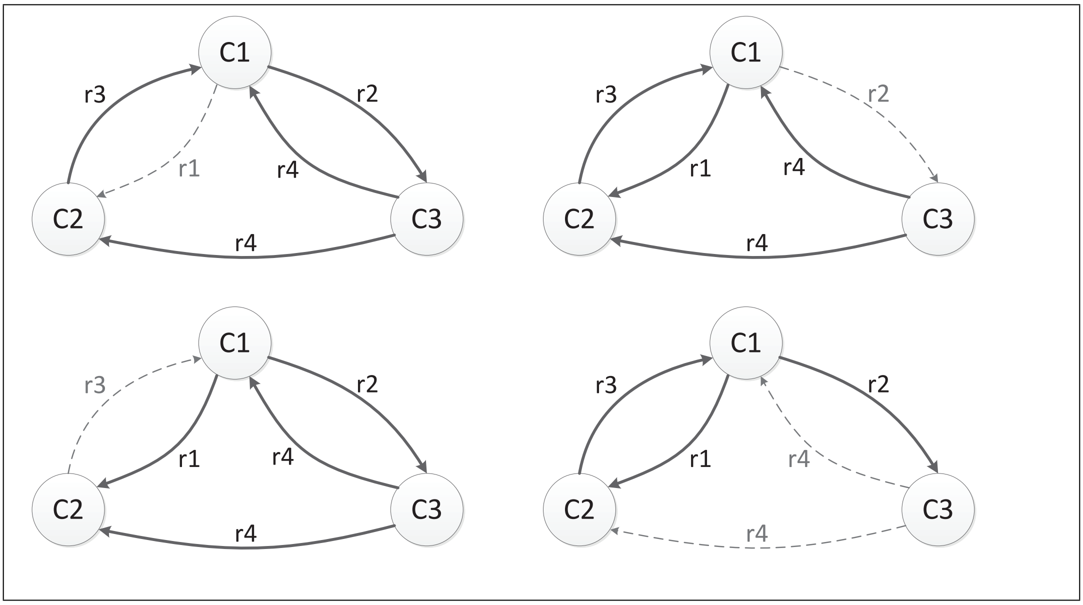

Figure 7: $DN(M(D_4))$

diagnosis is D_1. By applying D_1 the system will have the following equilibrium S':

$$\begin{aligned} S' = (&\{sensors, corba, distributedComputing\},\\ &\{profA, profB, middleware\},\\ &\{ubiquitousComputing, ambientComputing\}) \end{aligned}$$

6 Related Research

The present work takes as a starting point the notion of social dependence and dependence graphs introduced by Castelfranchi and colleagues [13, 28], and further developed with a more abstract representation, similar to ours, in Boella et al. [4] and in the context of the concept of conviviality defined as reciprocity, in Caire et al. [10, 11]. Dependence based coalition formation is analyzed by Sichman [27], while other approaches are developed in [26, 17, 3].

Similarly to Grossi and Turrini [20], our approach brings together coalitional theory and dependence theory in the study of social cooperation within multiagent systems. However, our approach differs as it does not hinge on agreements, and that we extend it to MCS.

In section 5, we referred to three alternative approaches for resolving inconsistencies in MCS. Two of them are based on local preferences [16, 2]. Our approach differs in that we take into account a global property of the system, conviviality, with the goal of maximizing its cooperativeness. Our solution can be combined with any of these approaches. For example, one may choose to apply the conviviality-based approach only to those diagnoses that comply with some constraints representing user-defined criteria, as proposed in [16]. Another solution would be to define hybrid criteria, which combine preferences on diagnoses, either if these are explicitly defined as in [16] or if they are derived from preferences on contexts as in [2], with conviviality-based criteria. A study of such combined approaches will be part of our future work.

Our solution is more similar to the approach of [15], which selects the subset-minimal diagnosis: for pairs $A = \{A_1, A_2\}$, $B = \{B_1, B_2\}$, the pointwise subset relation $A \subseteq B$ holds iff $A_1 \subseteq B_1$ and $A_1 \subseteq B_2$. Conviviality-based resolution subsumes this approach, since, by definition, between two diagnoses D_1 and D_2, for which it holds that $D_1 \subseteq D_2$, it will always select D_1. Additionally, as we showed in section 5, it can also select between diagnoses that cannot be compared using this relation.

7 Conclusion

Today, with the rise of systems in which knowledge is distributed in a network of interconnected heterogeneous and evolving knowledge resources, such as the Semantic Web, Linked Open Data, and Ambient Intelligence, research in contextual knowledge representation and reasoning has become particularly relevant. Multi-Context Systems (MCS) are logical formalizations of distributed context theories connected through a set of bridge rules, which enable information flow between contexts. The individual agents, which are represented as contexts, cooperate by sharing information through their bridge rules. By combining and reasoning on the information they import, they are able to derive new knowledge. Evaluating the ways in which the system enables cooperations, and characterizing a MCS based on the opportunities for information exchange that it provides are therefore, key issues. The social science concept of conviviality has recently been proposed to model and measure the potential cooperation among agents in multiagent systems and ambient intelligence systems. Furthermore, formal conviviality measures for dependence networks using a coalitional game theoretic framework, have been introduced. Roughly, more opportunities to work with other agents increase the conviviality of the system.

This paper is a step towards extending the concept of conviviality to MCS. First, we describe how conviviality can be used to model cooperation in MCS. Based on the intuition that agents depend on the information they receive from other agents to achieve their goals (e.g. to take more informed decisions), we define dependence networks for MCS. Furthermore, the aim is for MCS to be as cooperative as possible, and for agents to have as many choices as possible to cooperate with other agents. This results in MCS being as convivial as possible. In order to evaluate conviviality, we apply pairwise conviviality measures and allow for comparisons among different MCS. Finally we propose a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. In MCS, conflicts may arise as a result of importing mutually inconsistent knowledge from different contexts. Our approach is based on the idea that the optimal solution is the one that minimally decreases the conviviality of the system.

In further research, we contemplate the need to study alternative ways in which a MCS can be modeled as a dependence network. For example, another way to label dependencies among system contexts is to use the heads of the rules that these dependencies are derived from, instead of the rules themselves. This is based on the intuition that, the goal of applying a rule is actually to derive the conclusion that labels the head of the rule. This would require changing the definition of dependence networks to capture both disjunction (among rules that support the same conclusion) and conjunction (among the premises of each rule). We also plan to study the relation between the preference order on goals, which is included in the definition of dependence networks, and preferences on rules, contexts or diagnoses. Furthermore, we plan to combine the conviviality-based approach for inconsistency resolution with the preference-based approaches proposed by [16] and [2] and develop hybrid criteria for inconsistency resolution that take into account both local preferences and the conviviality of the system. Finally, we will study how the concept and tools for conviviality can be used in other distributed knowledge models, such as Linked Open Data, Distributed Description Logics [5], E-connections [22] and managed MCS [8], in which bridge rules are not only used to import information, but may also implement other operations, such as deletion or revision.

Acknowledgements

Thanks to the National Research Fund, Luxembourg (CO11/IS/1239572) CoPAInS project.

References

[1] Antoniou, G., Papatheodorou, C., Bikakis, A.: Reasoning about Context in Ambient Intelligence Environments: A Report from the Field. In: KR. pp. 557–559. AAAI Press (2010)

[2] Bikakis, A., Antoniou, G.: Defeasible Contextual Reasoning with Arguments in Ambient Intelligence. IEEE Trans. on Knowledge and Data Engineering 22(11), 1492–1506 (2010)

[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] Boella, G., Sauro, L., van der Torre, L.W.N.: Power and dependence relations in groups of agents. In: IAT. pp. 246–252. IEEE Computer Society (2004)

[5] Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. Journal of Data Semantics 1, 153–184 (2003)

[6] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: International Semantic Web Conference. pp. 164–179 (2003)

[7] Brewka, G., Eiter, T.: Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In: AAAI. pp. 385–390 (2007)

[8] Brewka, G., Eiter, T., Fink, M., Weinzierl, A.: Managed Multi-Context Systems. In: IJCAI. pp. 786–791 (2011)

[9] Brewka, G., Roelofsen, F., Serafini, L.: Contextual Default Reasoning. In: IJCAI. pp. 268–273 (2007)

[10] 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)

[11] 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)

[12] Caire, P., Bikakis, A.: Enhancing Cooperation in Distributed Information Systems Using Conviviality and Multi-Context Systems. In: Multi-disciplinary Trends in Artificial Intelligence - 5th International Workshop, MIWAI 2011, Hyderabad, India, December 7-9, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7080, pp. 14–25. Springer (2011)

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

[14] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, 2nd edn. (2001)

[15] 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)

[16] 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)

[17] Gerber, A., Klusch, M.: Forming dynamic coalitions of rational agents by use of the dcf-s scheme. In: AAMAS. pp. 994–995 (2003)

[18] Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=locality+compatibility. Artificial Intelligence 127(2), 221–259 (2001)

[19] Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. Artificial Intelligence 65(1) (1994)

[20] 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)

[21] Illich, I.: Tools for Conviviality. Marion Boyars Publishers, London (August 1974)

[22] Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: E-connections of abstract description systems. Artificial Intelligence 156(1), 1–73 (2004)

[23] 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)

[24] Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)

[25] 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)

[26] Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artif. Intell. 101(1-2), 165–200 (1998)

[27] Sichman, J.S.: Depint: Dependence-based coalition formation in an open multi-agent scenario. J. Artificial Societies and Social Simulation 1(2) (1998)

[28] Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: Procs. of The First Int. Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002. pp. 483–490. ACM (2002)

 Recieved January 2014