Introducing Conviviality as a property of Multi-Context Systems*

Antonis Bikakis[1] and Vasileios Efthymiou[2] and Patrice Caire[2] and Yves Le Traon[2]

[1] Department of Information Studies, University College London
a.bikakis@ucl.ac.uk
[2] University of Luxembourg, Interdisciplinary Center for Security, Reliability and Trust (SnT)
firstname.lastname@uni.lu

Abstract. Multi-Context Systems (MCS) are rule-based representation models for distributed, heterogeneous knowledge sources, called *contexts*, such as ambient intelligence devices and agents. Contexts interact with each other through the sharing of their local knowledge, or parts thereof, using so-called *bridge rules* to enable the cooperation among different contexts. On the other hand, the concept of *conviviality*, introduced as a social science concept for multiagent systems to highlight soft qualitative requirements like user friendliness of systems, was recently proposed to model and measure cooperation among agents in multiagent systems. In this paper, we introduce conviviality as a property to model and evaluate cooperation in MCS. We first introduce a formal model, then we propose conviviality measures, and finally we suggest an application consisting in a conviviality-driven method for inconsistency resolution.

1 Introduction

Multi-Context Systems (*MCS*) [1–3] 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 including 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 similar formal models of context including (*a*) the CYC common sense knowledge base [4], (*b*) contextualized ontology languages, such as Distributed Description Logics [5] and C-OWL [6], (*c*) context-based agent architectures [7, 8], and (*d*) distributed reasoning algorithms for Mobile Social Networks [9] and Ambient Intelligence systems [10].

While designing a real system based on the MCS model, comparing MCSs in order to select the most appropriate configuration, evaluations and measures are needed. For example, consider a house environment, which consists of various devices, sensors and appliances connected through a wireless network. The role of an Ambient Intelligence system is to transform this network of devices into a smart home environment by enabling devices to share and reason with their context knowledge. However, techniques

* Thanks to: National Research Fund, Luxembourg (I2R-SER-PFN-11COPA) CoPAInS project

to allow sharing of knowledge among the contexts could still be improved, to make Ambient Intelligence sustainable in the long term. A MCS may be used as the representation and reasoning model for such a system. *Bridge rules* are used to enable this sharing, by allowing each context to access the knowledge acquired by the other contexts. For example, consider two devices in our smart-house, that share their knowledge about the user's location within the house, to reason and optimize their service to this user's needs and desires. But, how can we then evaluate the ways in which the 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 introduce in MCS the notion of *conviviality*.

Defined by Illich as "individual freedom realized in personal interdependence" [11], conviviality has been 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 we interpret "freedom" as choice [12]. 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 [11]. We measure conviviality by counting the possible ways to cooperate, indicating degree of choice or freedom to engage in coalitions. Our coalitional theory is based on dependence networks [13, 14]; labeled directed graphs where the nodes are agents, and each labeled edge represents that the former agent depends on the latter one to achieve some goal. Here, we draw a parallel between, on the one hand an agent and a context, and on the other hand between a goal and a bridge rule. Particularly, 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 knowledge from other agents, as illustrated in Figure 1. The focus on dependence networks and more 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.

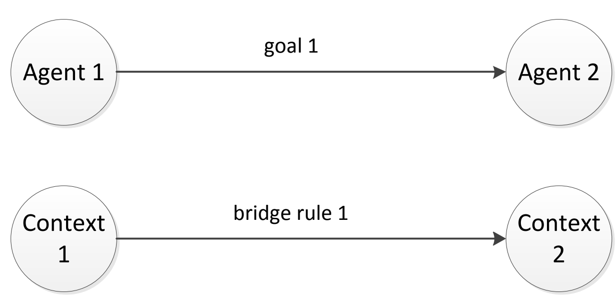

Fig. 1. The dependence network parallelism of contexts as agents, and bridge rules as goals. A labeled arrow, representing a goal, from a to b means that a depends on b to achieve this goal.

In distributed information systems, individual freedom is linked with the choice to keep personal knowledge and beliefs at the local level, while interdependence is understood as reciprocity, i.e. cooperation. Participating entities depend on each other to achieve the enrichment of their local knowledge.

Considering the potential applications of MCS and the notion of conviviality as described above, our main research question is the following:

How to introduce the concept of conviviality to Multi-Context Systems?

This 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?

Building on the ideas of [15], where we first identified ways in which conviviality tools, and specifically dependence networks and conviviality measures can be used to evaluate cooperation in Contextual Defeasible Logic, we propose:

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 (model and measures) for the problem of inconsistency resolution in MCS.

So far, most approaches for inconsistency (such as occurrence of $\alpha, \neg\alpha$) resolution in MCS have been based on the *invalidation* or *unconditional application* of a subset of the bridge rules that cause inconsistency [16–19]. They differ in the preference criterion that is applied for choosing among two or more candidate solutions. Here, we propose using the conviviality of the system as a preference criterion. This is 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 affects conviviality.

The rest of the paper is structured as follows: Section 2 presents formal definitions for MCS, as these were originally proposed in [3]. Section 3 proposes a model and measures for conviviality in MCS. Section 4 describes a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. Last section summarizes and presents directions for future work in the field.

2 Multi-Context Systems - Formal Definitions

For the needs of this paper we will use the definition of heterogeneous nonmonotonic MCS given in [3], according to which a MCS is a set of contexts, each composed of a knowledge base with an underlying logic, and a set of bridge rules which control the information flow between contexts. 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. We assume each element of $\mathbf{KB}_L$ is a set of "formulas".
– $\mathbf{BS}_L$ is the set of possible belief sets, where the elements of a belief set is a set of "formulas".
– $\mathbf{ACC}_L$: $\mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.

A *bridge rule* can add information to a context, depending on the belief sets which are accepted at other contexts. Let $L = (L_1, \ldots, L_n)$ be a sequence of logics. An L_k-bridge rule r over L is of the form

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

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

An *MCS* $M = (C_1, \ldots, C_n)$ is a collection of contexts $C_i = (L_i, kb_i, br_i)$, $1 \leq c_i \leq n$, 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 $kb_i \cup H \in \mathbf{KB}_{L_i}$, i.e., bridge rule heads are compatible with knowledge bases.

A *belief state* of an MCS $M = (C_1, \ldots, C_n)$ is a sequence $S = (S_1, \ldots, S_n)$ such that $S_i \in \mathbf{BS}_i$. 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 $j < l \leq m$: $p_l \notin S_{c_l}$. By $br_M = \bigcup_{i=1}^{n} br_i$ we denote the set of all bridge rules of M.

The above definitions are exemplified below. It is not in the scope of this paper to illustrate the many different logics that can be used in MCS. Furthermore, for the sake of clarity, our example is built on propositional logics only.

Example 1. Consider an MCS M, through which the software agents of three research students exchange information and classify research articles that they access in online databases. M contains contexts $C_1 - C_3$, each of which encodes the knowledge of each of the three agents. The knowledge bases for the three contexts are:

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

C_1 collects information about the keywords of the articles and encodes this information in propositional logic. In this case, the article under examination is about $sensors$ and $corba$ (Common Object Request Broker Architecture). C_1 also possesses the knowledge that centralized computing and distributed computing are two complementary concepts. C_2 uses propositional logic to encode additional information about articles, including the names of their authors; in this case $profA$ is the author of the article under examination. Finally, C_3 is an ontology of computing-related concepts, according to which $ubiquitousComputing$ is a type of $ambientComputing$.

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

$$r_1 = (1 : centralizedComputing) \leftarrow (2 : middleware)$$
$$r_2 = (1 : distributedComputing) \leftarrow (3 : ambientComputing)$$
$$r_3 = (2 : middleware) \leftarrow (1 : corba)$$
$$r_4 = (3 : ubiquitousComputing) \leftarrow (1 : sensors), (2 : profB)$$

Rule r_1 links the concept of $middleware$ used by C_2 to the concept of $centralized$-$Computing$ of C_1. r_2 expresses that $ambientComputing$ (a term used by C_3) implies $distributedComputing$ (a term used by C_1). r_3 expresses that $corba$ is a type of $middleware$, while r_4 expresses the belief of the third agent (C_3) that articles that are written by $profB$ and that contain $sensors$ among their keywords are about $ubiquitousComputing$.

Equilibrium semantics selects certain belief states of an MCS $M = (C_1, \ldots, C_n)$ as acceptable. Intuitively, an equilibrium is a belief state $S = (S_1, \ldots, S_n)$ where each context C_i respects all bridge rules 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 2. In the example given above, $S = (S_1, S_2, S_3)$ is the only equilibrium of the system:

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

S_3 is an empty set, since kb_3, the knowledge base of context C_3, is an empty set, $br_3 = \{r_4\}$, namely the set of bridge rules for context C_3 only consists of bridge rule r_4, and r_4 is not applicable in S, because $profB \notin S_2$.

3 Modelling and measuring conviviality in MCS

We mentioned in the introduction 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 the information dependencies among the contexts of a MCS and how conviviality measures can then be applied to measure conviviality in MCS. Our approach is based on the following ideas: (*a*) cooperation in MCS can be understood as information sharing among the contexts; (*b*) it is enabled by the bridge rules of the system; (*c*) therefore, bridge rules actually represent information dependencies among the contexts. Intuitively, that means conviviality will be captured through the different bridge rules that link the contexts.

3.1 Dependence Networks Model for MCS

According to [20], conviviality may be modeled by the reciprocity-based coalitions that can be formed. Some coalitions, however, provide more opportunities for their participants to cooperate with each other than others, being thereby more convivial. To represent the interdependencies among agents in the coalitions, [20] use dependence networks.

In this subsection, we first present Definition 1 from [20], which abstracts from tasks and plans. Then, building on [20]'s definition, we introduce our definition for a dependence network corresponding to a MCS.

A dependence network is defined by [20] as follows:

Definition 1 (Dependence networks). *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 *contexts* and *bridge rules*, we now introduce our definition, Definition 2, for a dependence network corresponding to a MCS, as follows:

Definition 2 (Dependence networks for MCS). *A dependence network corresponding to a MCS M, denoted as $DN(M)$, is a tuple $\langle C, R, dep, \geq \rangle$ where: C is the set of contexts in M, R is the set of bridge rules in M, $dep : C \times C \rightarrow 2^R$ is a function that is constructed as follows: for each bridge rule r (in the form of (1)) in R 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 $\geq: C \rightarrow 2^R \times 2^R$ is for each context a total pre-order on sets of bridge rules that the context appears in their heads.*

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 of the contexts c_i to achieve its goal, which is to apply r in order to infer s. It follows from Definition 2 that we can have two or more dependencies labeled by the same rule. The application of this rule relies upon all dependencies labeled with this rule. An alternative way to label dependencies would be 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, when using a rule, a context has actually the goal to derive the conclusion that labels the head of the rule. In that case, however, a new definition of dependence networks may be needed to support both conjunctions and disjunctions of dependencies.

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 [18, 10]. For sake of simplicity, we do not use this feature in the conviviality model that we describe below. 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 bridge rule r.

In our running example, the dependence network that corresponds to MCS M is the one depicted in Figure 2.

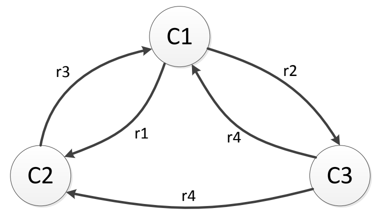

Fig. 2. The dependence network $DN(M)$ of MCS M of the running example. Nodes represent contexts and arrows represent dependencies. An arrow from context a to context b, labeled as r, means that a depends on b to apply bridge rule r.

In this graph, 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 one of the contexts C_1 and C_2, which appear in the body of r_4. This actually means that to apply rule r_4 in order to prove that the paper under examination is about ubiquitous computing, C_4 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.

3.2 Conviviality Measures

Conviviality measures have been introduced to compare the conviviality of multi agent systems [20], 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, [20] introduce formal conviviality measures for dependence networks using 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 we interpret freedom as choice, are stressed. Such measures provide insights into the type of properties that may be measured in convivial systems and thus reveal the quality of the system.

The conviviality measures presented in this work reflect the following Hypotheses:

- H1 the cycles identified in a dependence network are considered as coalitions. These 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 are implicitly signifying *simple cycles*, i.e., where all nodes are distinct [21]; we also discard self-loops. When referring to conviviality, we always refer to potential interaction not actual interaction.
- H2 conviviality in a dependence network is evaluated in a bounded domain, i.e., over a $[min, max]$ interval. This allows the comparison of different systems in terms of conviviality.
- H3 there is more conviviality in larger coalitions than in smaller ones.
- H4 the more coalitions in the dependence network, the higher the conviviality measure (ceteris paribus).

Hypothesis H1 is consistent with Definition 2, according to which each bridge rule is mapped to a set of potential dependencies between MCS contexts. The intuition for Hypothesis H3, is that a greater number of collaborating contexts in a MCS offers a greater source of knowledge. This means that each context participating in a large coalition has more available data, than the data it would have in a smaller coalition. Hypothesis H4 is motivated by the fact that a large number of coalitions indicates more interactions among contexts, which is positive in term of conviviality.

Our top goal is to maximize conviviality in the MCS. Some coalitions provide more opportunities for their participating contexts to cooperate than others, being thereby more convivial. Our two sub-goals (or Requirements) are thus:

R1 maximize the size of the contexts' coalitions, i.e. to maximize the number of contexts involved in the coalitions,
R2 maximize the number of these coalitions.

Following the definition of the *conviviality of a dependence network* [20], we define the *conviviality of a dependence network of a MCS M* as

$$\mathrm{Conv}(DN(M)) = \frac{\displaystyle\sum_{c_i,c_j\in C, i\neq j} \mathrm{coal}(c_i,c_j)}{\Omega}, \tag{2}$$

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

$$\Theta = \sum_{L=2}^{L=|C|} P(|C|-2, L-2)\times |R|^L, \tag{4}$$

where $|C|$ is the number of contexts in M, $|R|$ is the number of bridge rules in M, L is the cycle length, P is the 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 both c_i and c_j in $DN(M)$ and Ω denotes the maximal number of pairs of contexts in cycles (which produces the normalization mentioned in Hypothesis H2).

This way, the conviviality measurement of a dependence network, which is a rational number in [0,1], can be used to compare different dependence networks, with 0 being the conviviality of an acyclic dependence network and 1 the conviviality of a fully-connected dependence network.

Example 3. Following Equation 2 and the dependence network of M, which is graphically represented in Figure 2, we calculate the conviviality of $DN(M)$ of our running example, as:

$$\mathrm{Conv}(DN(M)) = \frac{10}{\Omega} = 0.208,$$

$$\text{where } \Omega = 480.$$

The result of Example 3 is just a way of comparing the conviviality of different systems. By itself it cannot be used to classify the conviviality of a MCS.

4 Use of conviviality as a property of MCS: Inconsistency Resolution

As we previously argued, conviviality is a property that characterizes the cooperativeness of a MCS, namely the alternative ways in which the contexts of a MCS can share information in order to derive new knowledge. By evaluating conviviality, the system may propose the different ways in which it can be increased, e.g. by suggesting new connections (bridge rules) between the system contexts.

Consider, for example, a MCS, in which a context does not import any information from other contexts. Recommending other contexts that this context could import information from would be a way to increase the conviviality of the system, which would in turn lead to enriching the local knowledge of the context but also the knowledge of the whole system.

4.1 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 an MCS, even if contexts are locally consistent, their bridge rules may render the whole system inconsistent. This is formally described in [3] 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 [16] as diagnosis:

"Given an 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 $M[br_M \backslash D_1 \cup heads(D_2)]$ 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. As it is obvious, in a MCS it is possible that there is more than one diagnosis that can be applied to restore consistency.

Example 4. 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, $ubiquitousComputing$ 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 choosing a diagnosis including:

- the number of bridge rules contained in the diagnosis - specifically in [16] subset-minimal diagnoses are preferred,
- local preferences on diagnoses proposed in [19], and
- local preferences on contexts and provenance information, which have been proposed for Contextual Defeasible Logic [18, 10].

4.2 Proposed Solution

Our approach is to use the conviviality of the resulted system as a criterion for choosing a diagnosis. This actually means that for each candidate solution (diagnosis), we measure the conviviality of the system that is derived after applying the diagnosis, and we choose the diagnosis that minimally decreases the conviviality of the system. The intuition behind this approach is that the system should remain as *cooperative* as possible, and this is achieved by enabling the maximum number of agents to both *contribute* to and *benefit* from this cooperation.

Diagnoses contain two types of changes that one can apply in the bridge rules: invalidation (removal) of a rule; and applying a bridge rule unconditionally, which actually 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 this is defined in (1)) in a MCS M, all the dependencies that are labeled by r are removed from the dependence network of M.

Assuming that $DN(M, D_i)$ is the dependence network that corresponds to MCS M after applying diagnosis D_1, the optimal diagnosis is the one that maximizes the conviviality of $DN(M, D_i)$:

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

Example 5. In the running example, there are four diagnoses that we can choose from: D_1-D_4. Each of them requires invalidating one of the four bridge rules r_1 to r_4, respectively. Figures 3 to 6 depict the four dependence networks $DN(M, D_i)$, which are derived after applying D_i. Dashed arrows in Figures 3-6 represent the dependencies that are dropped in each $DN(M, D_i)$, by applying diagnosis D_i.

Following Equation 2 and the four dependence networks, which are graphically represented in Figures 3-6, the conviviality of each $DN(M, D_i)$ is:

$$\mathrm{Conv}(DN(M, D_1)) = \frac{8}{\Omega} = 0.037 \text{ and}$$

$$\mathrm{Conv}(DN(M, D_2)) = \mathrm{Conv}(DN(M, D_3)) = \mathrm{Conv}(DN(M, D_4)) = \frac{2}{\Omega} = 0.009,$$

$$\text{with } \Omega = 216.$$

Since the number of goals $|G|$ is now 3, instead of 4, Ω has a different value than in $DN(M)$. By applying D_1 (Figure 3), only one cycle (C_1, C_2) is removed from the initial dependence network $DN(M)$, illustrated in Figure 2. However, by applying any

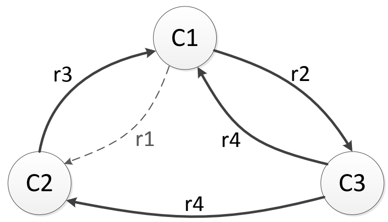

Fig. 3. $DN(M, D_1)$

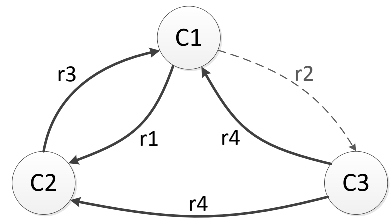

Fig. 4. $DN(M, D_2)$

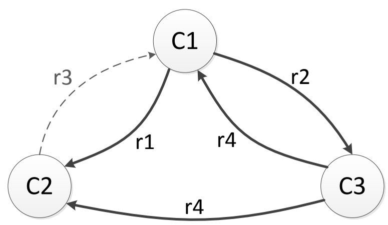

Fig. 5. $DN(M, D_3)$

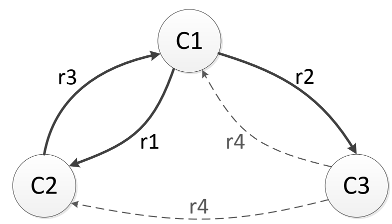

Fig. 6. $DN(M, D_4)$

of the remaining diagnoses D_2-D_4, two cycles are removed from $DN(M)$. Specifically, by applying D_2 (Figure 4), we remove the cycles (C_1, C_3) and (C_1, C_3, C_2). By applying D_3 (Figure 5), we remove the cycles (C_1, C_2) and (C_1, C_3, C_2). Finally, by applying D_4 (Figure 6), we remove the cycles (C_1, C_3) and (C_1, C_3, C_2).

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

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

This approach can also be combined with any of the approaches that have been proposed so far for inconsistency resolution. 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 suggested in [19]. It can also be combined with preferences on diagnoses proposed by [19] or preferences on contexts suggested by [18, 10]. A study of such combined approaches will be part of our future work in the field.

5 Conclusion

Today, with the rise of systems in which knowledge is distributed in a network of interconnected heterogeneous and evolving knowledge resources, such as 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 entities, that such systems consist of, cooperate by sharing information through their bridge rules. By 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 to its contexts 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. 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 toward extending the concept of conviviality, modeled with dependence networks, to Multi-Context Systems. First, we describe how conviviality can be used to model cooperation in MCS. Based on the intuition that contexts depend on the information they receive from other contexts to achieve their goals, i.e., apply specific bridge rules to infer particular information, we define dependence networks for MCS. Furthermore, the aim is for MCSs to be as cooperative as possible, and for contexts to have as many choices as possible to cooperate with other contexts. This results in MCS being as convivial as possible. In order to evaluate the conviviality of a MCS, we apply pairwise conviviality measures and allow for comparisons among MCS. Finally we propose a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. Indeed, without considering contextual information, reasoning can easily encounter inconsistency problems, for example, when considering knowledge in the wrong context. Our approach in this case 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 by labeling dependencies with the heads of the rules that they are derived from. 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 that the system contexts may have. Furthermore, we plan to combine the conviviality-based approach for inconsistency resolution with the preference-based approaches proposed by [19] and [18, 10]. Finally, we want to study how the concept of conviviality and the 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 [17].

References

1. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. Artificial Intelligence **65**(1) (1994)
2. Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=locality+compatibility. Artificial Intelligence **127**(2) (2001) 221–259
3. Brewka, G., Roelofsen, F., Serafini, L.: Contextual Default Reasoning. In: IJCAI. (2007) 268–273
4. 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)

5. Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. Journal of Data Semantics **1** (2003) 153–184
6. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: International Semantic Web Conference. (2003) 164–179
7. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation **8**(3) (1998) 261–292
8. Sabater, J., Sierra, C., Parsons, S., Jennings, N.R.: Engineering Executable Agents using Multi-context Systems. Journal of Logic and Computation **12**(3) (2002) 413–442
9. Antoniou, G., Papatheodorou, C., Bikakis, A.: Reasoning about Context in Ambient Intelligence Environments: A Report from the Field. In: KR, AAAI Press (2010) 557–559
10. Bikakis, A., Antoniou, G., Hassapis, P.: Strategies for contextual reasoning with conflicts in Ambient Intelligence. Knowledge and Information Systems **27**(1) (2011) 45–84
11. Illich, I.: Tools for Conviviality. Marion Boyars Publishers, London (1974)
12. 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. (2008) 1265–1268
13. Castelfranchi, C.: The micro-macro constitution of power. Protosociology **18** (2003) 208–269
14. 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, ACM (2002) 483–490
15. 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. Volume 7080 of Lecture Notes in Computer Science., Springer (2011) 14–25
16. 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)
17. Brewka, G., Eiter, T., Fink, M., Weinzierl, A.: Managed Multi-Context Systems. In: IJCAI. (2011) 786–791
18. Bikakis, A., Antoniou, G.: Defeasible Contextual Reasoning with Arguments in Ambient Intelligence. IEEE Trans. on Knowledge and Data Engineering **22**(11) (2010) 1492–1506
19. 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. Volume 6341 of Lecture Notes in Computer Science., Springer (2010) 143–155
20. 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)
21. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. The MIT Press (2001)
22. Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: E-connections of abstract description systems. Artificial Intelligence **156**(1) (2004) 1–73