International Journal of Approximate Reasoning 75 (2016) 39–56

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Preferential Multi-Context Systems

Kedian Mu [a,*], Kewen Wang [b], Lian Wen [b]

[a] *School of Mathematical Sciences, Peking University, Beijing 100871, PR China*
[b] *School of Information and Communication Technology, Nathan Campus, Griffith University, 170 Kessels Road, Nathan, Brisbane, Queensland 4111, Australia*

ARTICLE INFO

Article history:
Received 5 May 2015
Received in revised form 19 April 2016
Accepted 20 April 2016
Available online 22 April 2016

Keywords:
Multi-context systems
Equilibrium
Diagnosis
Inconsistency

ABSTRACT

Multi-Context Systems (MCSs) introduced by Brewka and Eiter are a promising way to interlink decentralized and heterogeneous knowledge contexts. In this paper, we propose Preferential Multi-Context Systems (PMCSs), which provide a framework for incorporating preference over contexts in MCSs. In such a PMCS, its contexts are divided into different strata according to the preference such that information flows only from a context to contexts in the same stratum or less preferred strata. Given a positive integer l, the first l preferred strata of a PMCS are able to fully capture information exchange between the contexts of these strata, and thus these contexts form a new PMCS called the l-section of the original PMCS. We generalize the equilibrium semantics for an MCS to $l_{\leq}$-equilibria for a PMCS. An $l_{\leq}$-equilibrium represents a belief state that is acceptable at least for the l-section of the given PMCS. We also investigate inconsistency analysis in PMCS and related computational complexity issues.

1. Introduction

Many (if not all) real-world applications of sharing and reasoning knowledge are characterized by heterogeneous contexts, especially with the advent of the world wide web. Research in representing contexts and information flow between contexts has gained much attention recently in artificial intelligence [3–5,9,7,12,18,8,20] as well as in applications such as software engineering and requirements engineering [15,22,24].

Instead of finding a universal knowledge representation for all contexts, it has been increasingly recognized that it is desirable to allow each context to choose a suitable representation tool for its own to capture its knowledge precisely. For example, in some frameworks such as Viewpoints for eliciting and analyzing software requirements, developers often encourage stakeholders to use their own familiar terms and notations to express their demands so as to elicit requirements as fully as possible [15,24]. Moreover, the heterogeneous nature of contexts may allow different monotonic or non-monotonic reasoning mechanisms to occur together in a given system. For example, as mentioned in [7], there is growing interest in combining ontologies based on description logics with non-monotonic formalisms in semantic web applications. However, the diversity of representations of contexts in such cases brings some important challenges to accessing each individual context as well as to interlinking these contexts [8].

Nonmonotonic Multi-Context Systems (MCSs) introduced by Brewka and Eiter [7] are a promising way to represent and reason about heterogeneous and nonmonotonic contexts [8]. Instead of attempting to translate all contexts with different

* Corresponding author.
E-mail address: mukedian@math.pku.edu.cn (K. Mu).

http://dx.doi.org/10.1016/j.ijar.2016.04.006

formalisms into a unifying formalism, they leave the logics of contexts untouched and interlink contexts by modeling the inter-contextual information exchange in a uniform way. To be more precise, information flow among contexts is articulated by so-called bridge rules in a declarative way. Similar to logic programming rules, each bridge rule consists of two parts, the head of the rule and the body of the rule (possibly empty). More importantly, each bridge rule allows access to other contexts in its body. This makes it capable of adding information represented by its head to a context by exchanging information with other contexts. The semantics of Multi-Context Systems is defined in terms of equilibria that are acceptable belief states [7].

Multi-Context Systems can be viewed as the first step towards interlinking distributed and heterogeneous contexts effectively. The way they operate on contextual knowledge bases is only limited to adding information to a context when the corresponding bridge rules are applicable [8]. To be more applicable to real-world applications, it is advisable to generalize Multi-Context Systems from some perspectives. For example, Brewka et al. have considerably generalized Multi-Context Systems to Managed Multi-Context Systems (mMCS) by allowing flexible operations on context knowledge bases [8]. Essentially, Managed Multi-Context Systems focus on managed contexts, which are contexts together with possible operations on them.

The problem of combining preferences and contexts has been raised by Brewka [6]. In practical applications, it is often the case that some context has higher priority over another context. In software engineering, prioritizing stakeholders and their demands is considered as a practical strategy for developing a software system with limited resources [22,21]. Moreover, preference relations on contexts of a system are considered as useful information to make some trade-off decisions on repairing the system in the presence of inconsistency in many real-world applications. For example, in the prioritized viewpoints framework, the priority levels of viewpoints play a dominant role in resolving conflicts between software requirements belonging to different viewpoints [22,21].

On the other hand, preference relations on contexts can have a significant impact on information exchange between contexts and inter-contextual knowledge integration in real-world applications. For example, it is intuitive to revise a less reliable knowledge base by accessing more reliable ones. But we cannot use information deriving from less reliable sources to revise more reliable knowledge bases in many cases. In legal reasoning, consequences of applying a law to a case can be rebutted by that of applying another law with higher level when there is a conflict, and not vice versa. In such cases, it may be advisable to take into account preferences on contexts in characterizing inter-contextual information exchange in Multi-Context Systems.

Instead of regarding a whole system useless in the presence of inconsistency, some consistent subsystems are still of interest to a prioritized system when the cost of repairing the whole system is too expensive in practical applications. Such a subsystem is often required to have some appropriate balance between consistency and the preference of contexts excluded from that subsystem. For example, in many cases in a multi-party negotiation, an agreement between the most important parties is preferred if it is difficult to achieve an agreement between all parties. In incremental software development, only requirements with priorities higher than a given level are concerns of developers at a given stage.

Bikakis et al. [5] proposed an approach for handling preference in MCSs. In their framework, each context is a theory in defeasible logic and the information flow between contexts is also implemented through a mechanism similar to the procedure for defeasible logic. In their approach, every context identifies those defeasible bridge rules that need to be ignored in the purpose of avoiding conflicts based on the preference. However, this approach is only defined for MCSs whose contexts are theories of defeasible logic. It is unclear how their approach can be generalized.

In this paper, we propose a framework for incorporating preference information in Multi-Context Systems, called *Preferential Multi-Context Systems* (or *PMCSs*). In syntax, a PMCS is given in the form of a sequence of sets of contexts such that the location of a set signifies its preference level. Without loss of generality, we assume that the smaller the location of a set is, the more preferred contexts in that set are. We call each set of contexts in that sequence *a stratum*. Then an MCS can be regarded as a special PMCS that has only one stratum. As explained above, we require that information flow can only be from strata with higher priority to strata with lower priority in a PMCS. That is, any bridge rule of a given context is not allowed to access other strictly less preferred contexts in its body. As such, the first few strata capture all information exchange between the contexts in these strata, and then also compose a new PMCS such that all the contexts involved in it are strictly more preferred than the rest in the original PMCS. Such a sub-PMCS is referred to as a *section* of that system. We are interested in its sections as well as the whole PMCS and thus, the notion of $l_{\leq}$*-equilibria* is proposed to represent belief sets acceptable for at least contexts in the first l strata for a positive integer l.

Since a PMCS is distributed and heterogeneous, it should have the capability of handling inconsistency. So we take into account the impact of the preference relation over contexts on inconsistency analysis in a PMCS. The notion of diagnoses for standard MCS [12,13] can be easily adapted to PMCS. However, the availability of preference information allows us to develop a more efficient and effective approach to handling inconsistency in PMCSs. In some practical applications of recovering consistency such as in requirements engineering, instead of fixing the whole system, certain least preferred information is simply discarded if the cost of fixing actions is too expensive. This observation leads us to consider the largest section of a given PMCS that has equilibria, called the *maximum consistent section*. It is important since it fully captures the information exchange between contexts in a reliable part with the largest number of strata of the PMCS. As a result, we favor those diagnoses that do not touch the maximum consistent section, and thus introduce the notion of compatible diagnoses (i.e., those compatible with the maximum consistent section). The refined notion is more useful for handling inconsistency of PMCSs. The compatibility relies on availability of the maximum consistent section but we note that if the given PMCS has m strata, then at most m checks of consistency are needed in order to find the maximum consistent