

Ontologies as Contexts for Constraint-Based Reasoning

Richard J. Wallace, Tomas E. Nordlander, and Ioannis Dokas

Cork Constraint Computation Centre and Department of Computer Science
University College Cork, Cork, Ireland
email: {r.wallace,t.nordlander,i.dokas}@4c.ucc.ie

Abstract. We give a brief description of a system we are currently building that fulfils the promise of our title. This description also indicates why we are interested in the topic of contexts and ontologies.

1 Introduction

Many problems involve finding a solution subject to certain constraints. Constraint programming is an important technique for modelling such problems, which supports combinations of inference and search that allow more efficient solving. However, this cleverness is obtained at a cost, since these methods normally required highly skilled specialists as well as a lengthy period of time for modelling and crafting of solvers.

The question we are interested in here is whether these methods can be embedded so that a layperson using such a system can make use of these methods without having to deal with them directly. In this case it is necessary that a constraint model - with at least a simple form - be built automatically, and that the solving process proceed without the user being aware of the particular kind of representation involved. What the user is aware of is the original problem, stated in everyday terms that he or she is familiar with.

In order to bring about this result, we are interesting in using ontologies to represent the problem in a simple, qualitative manner, which can also serve as the basis for deducing a more abstruse CSP model. The ontology, therefore, provides the context for constraint modelling and constraint-based reasoning. (In fact, it can provide many contexts as the following section will show.)

2 Description of an Ontology-and-Constraint Based System

We are presently trying to implement our ideas in a product selection system called the Matchmaker. Product selection is carried out using a form of interaction in which the system presents individual products and the user critiques them. (This is called a “suggestion/correction cycle”.) Internally, the problem of product selection is represented as a constraint satisfaction problem, but with each critique the representation is updated to conform to the additional constraints revealed on this cycle. (This scheme is presented in more detail in [1].)

The Matchmaker program is written in Java, and Protege-OWL is being used to build the product ontology. In its final form, the system will be web-based and the system therefore falls under the heading of e-commerce.

Conceptually, a product is not only an item of interest but is also possibly an object, an artefact, an organisation, etc., and is associated with some type of setting, or even a specific location. In order to carry out the matchmaking process in a ‘knowledgeable’ manner, a product must be viewed in all of these ways. At the same time, there are restrictions on the degree of detail or the level of technical sophistication that can be handled efficiently; hopefully, there are also limits on what the user needs to know in order to make a satisfactory selection.

The resulting conceptualisation of a product can be thought of as a set of contexts in which the item in question is embedded. A straightforward and efficient way to represent these various contexts is through an ontology. In this case, if we start with a particular product type, we can reach associated concepts by means of a class hierarchy of the usual type, as well as deducing relevant relations (“properties” in the usual parlance) involving these classes. Our expectation is that such context-supported interaction will help us achieve the kind of interaction envisaged for the Matchmaker system in a sufficiently compelling fashion.

3 Ontologies to CSPs

Another major challenge necessitated by our approach is connecting the ontology to the constraint solver in a way that supports the computational requirements, while being transparent to the user. This is related to another motivation for using ontologies, which is to allow users to work with familiar concepts, without having to think in terms of the rather abstruse language of constraint programming and constraint satisfaction. In the implementation we are building, this is done by translating relevant ontology classes into variables, following rules that support the CSP formalism, in particular, the rule of single assignments to a variable.

An alternative approach, that we will consider in the future, is to use a CSP ontology to support the mapping between subject-matter concepts and the CSP representation. Some work along these lines has been done by other authors [2].

4 Future Vistas

The use of ontologies to provide conceptual contexts for constraint solving and to ‘shield’ users from constraint programming details is a general idea that may be useful in a number of application areas. Obvious candidates are systems for scheduling and configuration.

References

1. Freuder, E.C., Wallace, R.J.: Suggestion strategies for constraint-based matchmaker agents. *International Journal on Artificial Intelligence Tools* **11** (2002) 3–18
2. Sleeman, D., Chalmers, S.: Assisting domain experts to formulate and solve constraint satisfaction problems. In: *Proc. 15th Internat. Conf. on Knowledge Engineering and Knowledge Management, LNCS No. 4248, Berlin, Springer (2006)* 27–34