Dynamic Ontology Refinement

Fiona McNeil, Alan Bundy, Chris Walton

f.j.mcneill@ed.ac.uk,{bundy,cdw}@inf.ed.ac.uk

Abstract: The development of the semantic web makes the facilitation of agent communication an issue of increasing importanc
is usually assumed that agents are using the same ontology and hence can understand one another, but the dynamic and distrib
nature of the semantic web mean that this is not always a valid assumption. We describe a system that can dynamically disc
ontological mismatches between agents during communication and then refine them, to enable communication between these ag

Introduction

We are exploring the situation in which a *plan-implementation* (PI) agent is attempting to achieve goals by forming and executing plans in a given domain. Each plan step will be executed through interaction with other agents: for example, a plan step `(Buy Ticket)` may be executed through locating and communicating with a *ticket-selling agent*.

Ideally, all the agents with which the PI interacts will have the same ontology as it and hence the plan execution will proceed satisfactorily. However, because ontologies are dynamic and can be adapted for particular tasks, other agents may have ontologies that are similar but are slightly out of sync. This can lead to plan execution failure.

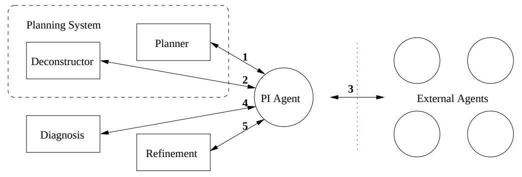
There are two main types of error that lead to failure: errors of representational adequacy and errors of fact. These correspond to two different parts of the PI agent's ontology:

- the *signature*, which describes the types of entities in the domain;
- the *theory*, which describes the instantiations of the signature.

The ontology mismatch is highlighted by a particular error in the theory, which will usually indicate an error in the underlying signature. Once this signa-

of the task, but differs from the main focus of the project, which centres on refining ontologies syntactically (signature refinement) rather than purely semantically (theory refinement). The main difference with the latter is that ontology mapping usually assumes that one has complete access to the ontologies one is attempting to map, whereas we are attempting this with the partial information that can be extracted from plan failure and from putting specific questions to other agents.

Refinement System



System Architecture

The flow of execution in the system is as follows:

1. The PI agent sends its ontology, together with the goal, to a planner.
2. The planner produces a plan for achieving the goal which consists of a list of actions. This is sent, together with the original representation of the ontology, to a *plan-deconstructor*. The purpose of this

communication is then used to diagnose the precise ontology mismatch (see below).

4. The diagnosis and information about how it should be refined is passed to the refinement system. This system corrects the error and passes the corrected part of the ontology back to the PI agent, which updates its ontology accordingly.
5. The process now begins again, with the PI agent forming plans on the basis of a more reliable ontology.

The cycle will continue until the ontology is sufficiently accurate for the goal to be achieved.

Diagnosing the Problem

When the plan fails, the justification for the plan step is referred to. This justification will provide a rule and a justification for why each of the preconditions are believed to be true, any of which may be incorrect.

Much information can be obtained by observing the point of failure. For example, did failure occur immediately after a request was made to another agent? Or did the other agent put some queries to the PI? If there were queries, were any of them surprising? The PI would expect to be asked about some of its preconditions, so that the other agent could verify that they were correct before performing the action. However, the PI would not expect to be asked