Integration of learning into a knowledge modelling framework

Josep Lluis Arcos                  Enric Plaza

Artificial Intelligence Research Institute, IIIA.
Spanish Council for Scientific Research, CSIC.
Camí de Santa Bàrbara, 17300 Blanes, Catalunya, Spain.
{arcos | plaza}@ceab.es

Abstract. In this paper we will report our current research on the NOOS language, an attempt to provide a uniform representation framework for inference and learning components supporting flexible and multiple combination of these components. Rather than a specific combination of learning methods, we are interested in an architecture adaptable to different domains where multiple learning strategies (combinations of learning methods) can be programmed. Our approach derives from the knowledge modelling frameworks developed for the design and construction of KBSs based on the task/method decomposition principle and the analysis of knowledge requirements for methods. Our thesis is that learning methods are methods with introspection capabilities that can be also analyzed in the same task/method decomposition. In order to infer new decisions from the results and behavior of other inference processes, those results and behavior have to be represented and stored in the memory for the learning method to be able to work with them.

1 Introduction

One of the key issues in the current development of knowledge-based systems (KBS) is the degree to which different components can be combined in a seamless way. Specifically, the integration of learning components, both for knowledge acquisition and for learning from experience, is viewed as an essential topic for future KBS building and design.

We have developed NOOS, a representation language for integrating inference and learning components in an uniform representation. Our approach derives from the knowledge-level analysis of expert systems and the knowledge modelling frameworks developed for the design and construction of KBS. These knowledge modelling frameworks like KADS [20] or components of expertise [15] are based on the task/method decomposition principle and the analysis of knowledge requirements for methods. Our thesis is that learning methods are methods with introspection capabilities that can be also analyzed in the same task/method decomposition. Thus, learning methods can be uniformly integrated into our framework and represented as methods in the NOOS language.

Although this will be explained in the paper, the main idea is that any time some
knowledge is required by a problem solving method, and that knowledge is not directly available there is an opportunity for learning. We call those opportunities impasses, following Soar terminology [8], and the integration of learning methods is realized by methods that solve these impasses. A innovation required for integrating learning components into knowledge modelling frameworks is the definition of two levels: domain level and inference level. Domain level is where domain knowledge is modelled usually in knowledge modelling frameworks and is used to solve domain problems. Inference level is a level where domain knowledge shortcomings (impasses) are detected and learning components to overcome them can be defined. The inference level is thus a meta-level that deals with problems arising from the base-level problem-solving process. As we will see in section 3, different approaches to overcome an impasse are possible to be programmed into our system and integrated seamlessly. The second innovation to integrate learning into knowledge modelling frameworks is the notion of memory: memory of successes and failures of methods into solving tasks is necessary, as shown in section 2. This allows us to integrate case-based (analogical) reasoning and inductive learning methods in our framework, since the results of past problem solving cases are stored in the system’s memory.

Next section describes the elements of our knowledge modelling framework and their implementation in the NOOS language. Section 3 shows an example where multiple learning components are applied to achieve the knowledge requirements of a (simplified) diagnosis task. Related work is compared in section 4 and, finally, section 5 discusses our approach and our future work.

2 The knowledge modelling framework

The elements that build up our framework are tasks, methods, theories and case models. This framework is close to the components of expertise [15], however our implementation of it, the NOOS language, departed from the original definition, so we do not claim we are using that componential framework. Our framework augments the ideas of the components of expertise with the notion of episodic memory: the memorization of problem-solving episodes allows learning methods to be integrated since they require to access the past experience to improve the system performance.

Let us shortly describe the elements of our knowledge modelling framework. Tasks are goals to be achieved by the system in a problem setting. Problem solving methods are specifications of ways to achieve tasks. Usually, tasks are decomposed into subtasks by means of a problem-solving method, e.g. the generate-and-test method applied to a task decomposes it into the generate and test subtasks. This recursive decomposition of task into subtasks by means of a method is called the task/method decomposition. This recursive task/method decomposition finishes when a task uses an elementary method (like union and intersection of sets) provided by NOOS language. Case models and theories embody the domain knowledge modelled for a given problem. A case model contains all factual knowledge of a topic and consists of the set of tasks that make sense for it. For instance, the case model of John are those tasks we have solved about John (like his fever being 39) and those tasks to be solved (like finding his diagnosis). Thus solving a problem consists on completing the case model of the problem (e.g. finding the diagnosis of John) by means of a method (e.g. generate-and-test). As the case model is