From: Proceedings of the Eleventh International FLAIRS Conference. Copyright © 1998, AAAI (www.aaai.org). All rights reserved.

A New Model of Reflective Introspective Learning

Susan Fox
Department of Mathematics and Computer Science
Macalester College
1600 Grand Avenue
Saint Paul, MN 55105 USA
E-mail: fox@macalester.edu

Abstract

Systems which introspect about their own processes can improve their reasoning behavior in response to experience using "introspective learning" techniques. Many systems which perform introspective learning analyze and change only an underlying domain task's reasoning processes. They do not possess the ability to reflectively introspect about the introspective task itself. We present a model of a truly reflective introspective learning system which uses the same case-based reasoning mechanisms for both its domain and introspective tasks. The reuse of case-based reasoning mechanisms enables the reuse of introspective models and mechanisms developed for a CBR domain reasoner to reflect about the introspective process itself. We suggest an introspective model capable of reflection, and begin to address the issue of balancing introspection and action within a reflective system.

Introduction

Meta-reasoning — reasoning that incorporates knowledge of the task being addressed and methods for achieving it — has been used to predict the behavior of other agents (e.g., (Gurer, des Jardins, & Schlager 1995)), to guide the acquisition and application of domain knowledge (Bradzil & Konolige 1990; Clancy 1988; Davis 1982), and to adjust the system's own reasoning processes in response to feedback about its performance in its domain (Collins *et al.* 1993; Ram & Cox 1994; Cox & Freed 1995; Stroulia & Goel 1995). This last application of meta-reasoning has been referred to as "introspective reasoning" or "introspective learning."

Most work on introspective reasoning focuses on improving the performance of an underlying reasoning system on some domain task. Introspective reasoning is rarely applied to the introspective reasoning process itself; the system is never fully "reflective." A reflective system must be able to manipulate and reason about its own processes, and alter its own processing behavior (Ibrahim 1992). SOAR (Rosenbloom, Laird, & Newell 1993b) can reflectively reason about the constructs it makes, but does not explicitly represent its introspective model of behavior and cannot necessarily introspect about all of its mechanisms.

Incorporating reflection into an introspective learning framework requires extending the system's knowledge about its reasoning processes to include the learner itself. Unlike a system where introspection is only applied to an underlying task, a reflective introspective learner must control when and to what extent introspection occurs, so that it does not choose to spend all its time constructing a reflective tower.

In this paper we present a preliminary model for reflective introspective learning, using case-based reasoning (CBR) techniques to implement both domain and introspective processes. We explore the needs of such a system for representing meta-cognitive reasoning processes, and begin to address the balance between taking action and introspecting, and the costs associated with unrestricted reflection.

Our initial results suggest that the use of case-based reasoning for introspective learning carries little overhead compared to other methods. Monitoring of the introspective process is feasible and requires little additions to an existing model of case-based processes. The correct balance between reflecting and acting on the domain task is still an open question, but we suggest that reflective introspection be driven by necessity only. By default, introspection should occur only at the basic level, and reflections to higher levels should occur only in failure situations when lower level introspection has failed to resolve the problem.

Reflection

A system is reflective if it can shift the focus of its processes from its basic "domain" task to the problem-solving task itself (Smith 1982). Such a system can construct an unbounded "reflective tower" of reasoning processes where each analyzes the process beneath it. The idea of reflection has been applied to programming language designs as well as artificial intelligence reasoning systems (Ibrahim 1992).

Ideally, a reflective system should use the same reasoning mechanisms at all levels of operation. It can reason about its own processes, including the introspec-

Copyright © 1998, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

tive processes themselves. This ability to reason at any required level of abstraction should permit a reflective system to adapt to its environment flexibly: it can introspectively consider and alter any aspect of its reasoning in response to its experiences.

Reflection used for introspective learning must control the proliferation of reflected reasoning level. Introspecting about the domain task could trigger introspecting about the introspection task, which could then trigger further introspection at higher levels. Many reflective systems take an "as-needed" or failure-based approach to moving to a higher level of abstraction: this is the approach we have taken with RILS[1].

In the next section we discuss a range of systems performing introspective learning. To our knowledge, no systems exist which permit reflective introspective learning and which maintain an explicit model of ideal behavior against which to judge their reasoning processes. That is the goal for RILS.

Background on Introspective Reasoning

In recent years a number of different approaches to introspective reasoning have been explored. The focus has been detecting opportunities to adjust a system's reasoning process, and diagnosing reasoning failures to determine what adjustment to make.

Meta-AQUA maintains reasoning trace templates (Meta-XPs) which describe the patterns of reasoning that indicate reasoning failures (Cox 1996; Ram & Cox 1994). In theory Meta-AQUA's Meta-XPs could be applied to the introspective process itself, but reflection was not the focus of the project.

Autognostic uses a "Structure-Behavior-Function" model to represent reasoning processes (Stroulia 1994). RAPTER (Freed & Collins 1994) and CASTLE (Collins *et al.* 1993; Krulwich, Birnbaum, & Collins 1992) use model-based reasoning: an explicit model of ideal reasoning behavior is examined to diagnose failures. The model consists of "assertions" describing the desired behavior at each given point: for example, "The solution generated by adaptation will match the current situation." These systems do not include reflective capabilities.

IULIAN is an introspective reasoning system that does address reflection (Oehlmann, Edwards, & Sleeman 1994). Introspective reasoning is integrated with the overall domain task of IULIAN. It uses case-based planning to generate both domain plans and introspective plans. IULIAN is reflective, as introspective plans may apply equally to domain problems or introspective ones. However, its introspective knowledge is mostly implicit in introspective plans, and those plans have incomplete access to the mechanisms which use them.

The Massive Memory Architecture is a unified architecture for performing introspective reasoning and case-based reasoning (Arcos & Plaza 1993). Introspection is task-driven, and based on detection of impasses: tasks which have no known solution generate meta-tasks which search for a solution method. The MMA does not have an explicit model of ideal behavior. It cannot detect or learn from sub-optimal behavior, only catastrophic failures

Most of the previous systems use case-based reasoning to implement introspective reasoning. As a different approach, SOAR is a rule-based system which does deliberately address reflection (Rosenbloom, Laird, & Newell 1993b). SOAR's rule base contains rules which control the rule selection process itself and the "focus of attention" processes (Rosenbloom, Laird, & Newell 1993a). This permits SOAR to learn new behaviors by creating new meta-rules. Because the rules affect their own processes, SOAR can behave reflectively. SOAR does not include any explicit processes for analyzing its own behavior, and finding rules and applying rules are processes beyond the scope of its reflective control.

The ROBBIE system (Fox & Leake 1995; Fox 1995) drew original inspiration from a proposal by Birnbaum *et al.* (1991) to apply model-based introspective reasoning to CBR. Like RAPTER and CASTLE, its model is a collection of assertions describing the ideal reasoning process. ROBBIE incorporates detection of reasoning failures into the introspective system itself, whereas in RAPTER and CASTLE failure detection is performed by the underlying reasoning system. ROBBIE is not reflective; its introspective model describes only its underlying planning system. However, ROBBIE is a good starting point for creation of a reflective introspective reasoner that retains an explicit process model. In ROBBIE the same CBR mechanisms are used to implement both the primary planning task and a variety of subtasks, such as selection of adaptation strategies. A *case-based* introspective learner can use much of the existing introspective model to reflectively examine its own behavior.

Proposed Reflective Introspection

We propose a model of introspective learning that is truly reflective, and that includes an explicit, declarative model of the overall reasoning process. RILS uses case-based reasoning for all its tasks: domain and introspective. We leverage the creation of a reflective introspective learner by reusing existing CBR mechanisms and reusing and extending an existing introspective model of the case-based reasoning process.

The ROBBIE system includes a complete model of its CBR process applied to its domain task of route planning and executing. RILS adapts ROBBIE's introspective framework and model to become a case-based introspective learner. RILS represents its introspective knowledge as assertion cases. Each assertion case contains an assertion about some portion of the reasoning process, links to other causally-related assertions, possible repair strategies to correct a failure of that assertion, and statistics about the application of the assertion case and its outcome. Assertions describe expectations the system holds about the ideal behavior of a portion of

[1] Reflective Introspective Learning System

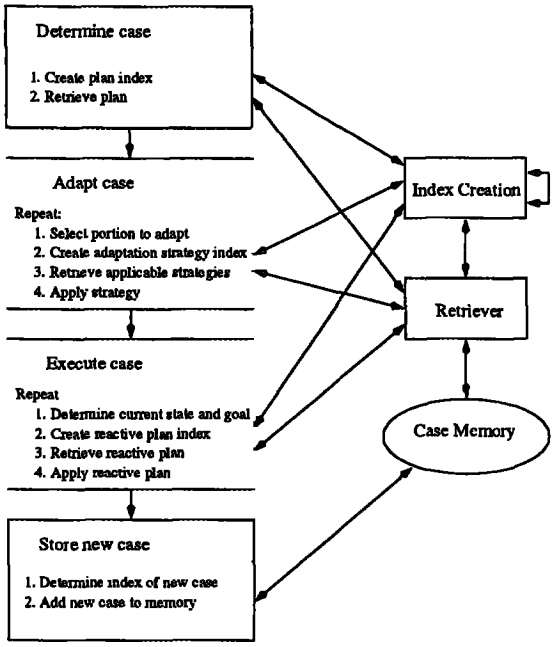

Figure 1: Domain level processes in RILS

the reasoning process. Assertions are chosen for introspective tasks by case-based retrieval; future work will extend this to include adaptation of assertion cases.

RILS uses the same retrieval process to retrieve domain task cases as introspective assertion cases. The assertions it has about domain-level case retrieval can, therefore, apply directly to its introspective task as well. Besides the assertions being reused from the ROBBIE system, RILS contains new assertions which explicitly describe the rest of the introspective learning process.

RILS' Introspective Reasoning Process

In order to understand the introspective reasoning task which RILS faces, it is necessary to understand the mechanisms used and reused within RILS. Figure 1 shows the domain task processes, and their relationship to the index creation and retrieval components. The introspective model breaks the case-based process into five components: index creation* using stored rules to elaborate indices; case retrieval; adaptation*, using stored adaptation strategies; execution*, using stored reactive action plans; and plan storage. The starred components are case-based processes themselves which reuse the mechanisms for index creation and retrieval.

The introspective model in RILS, represented as a collection of assertion cases, applies to both the domain task and the introspective task. RILS' introspective processes shown in Figure 2, are clearly heavily dependent on the same case-based processes as the underlying reasoning system.

Introspective learning in RILS involves three main phases: monitoring for reasoning failures, diagnosing a reasoning failure once detected, and repairing the underlying system. All three phases contain prominent case-based components: additional assertions have been added to the introspective model for other portions of the introspective process, but the same mechanisms for detecting and repairing reasoning failures apply.

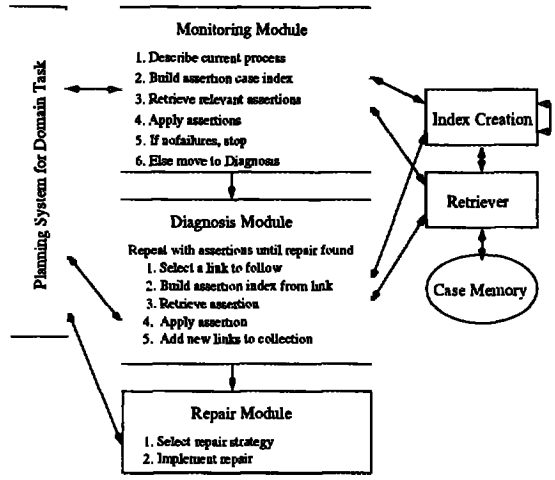

Figure 2: Introspective processes in RILS

While a reasoning process is underway, the monitoring component of RILS' introspective learner compares the actual behavior to assertions about ideal behavior. Assertion cases are retrieved from the case memory based on the component of the reasoning process being examined. Monitoring the introspective task is exactly the same process; though steps must be taken to ensure that monitoring the introspective process does not automatically generate unlimited reflective layers.

When an assertion failure is discovered, the diagnosis component begins a heuristic-guided search for a "root cause" for the failure: a prior, causally-related assertion whose failure led to the detected failure. The diagnosis component uses information in each assertion case about causal relationships with other assertions to retrieve related assertions for consideration. The ultimate goal is to find a related failed assertion which has a repair recommended for it.

Once a "root cause" has been determined, the repair component chooses a repair to the system and implements it. Control then returns to the reasoning process that was underway when the failure was detected. In the long run, the repair module itself should become a case-based process in which repair "recipes" are stored in the case memory.

Currently, RILS leverages off of ROBBIE's strengths again: repairs are restricted to learning new indexing features for case retrieval. In general any conceivable alteration to the system's reasoning process would be possible. Introspective learning in RILS will eventually include altering the weighting of criteria during retrieval and creating new adaptation strategies. Given the central importance of case retrieval to RILS, learning case indices is a powerful method.

```
(diagnose-spec2
    (assertion diagnosis specific 2 during)
    (and (contains-part assert-case links)
         (member-of-structure
             (part-value assert-case links)
             checked-assertions))
    (variables assert-case checked-assertions)
    (links (abstr (diagnosis general 2))
           (prev (diagnosis specific 1))
           (next (diagnosis specific 3)))
    (repair)
    (statistics (uses 12)
                (failures 0)))
```

Figure 3: An assertion case for the diagnosis component: "Every retrieved assertion will have a link to one already under consideration"

A Reflective Introspective Model

The introspective model of ideal behavior includes the assertions described in the ROBBIE system's model. In the ROBBIE system the model is a separate, monolithic data object. RILS' model is broken into assertion cases stored independently in its case memory: the context in which an assertion applies is the index of the assertion case. The assertions developed for the domain tasks are easily reused for a case-based introspective learner, and additional assertion cases describe other portions of the introspective reasoning process. These assertions are described using the same vocabulary as used for the case-based processes; ROBBIE's assertion vocabulary was designed to be generally applicable to a wide range of reasoning processes.

An example of an assertion case is given in Figure 3. This assertion describes an expectation about the diagnosis component of the introspective reasoner. The assertion states that every assertion case retrieved during diagnosis will contain a causal link to an assertion already under consideration.

The resulting model, distributed among the assertion cases, describes a single unified system which uses case-based retrieval for most of its informational needs. It has specialized assertions for the ways in which the different CBR components use the cases they retrieve.

We anticipate that RILS will be able to learn new indexing features for assertion cases as well as cases for its domain task. The new features would permit a more focused retrieval of assertions, optimizing the introspective learning process.

Balancing Reflection and Action

We have only begun to consider the issues surrounding reflective introspective learning. We have experimented with two initial modes for RILS: introspective learning only at the domain level, by default; and introspective monitoring of the introspective task itself at its first level. In both cases, further reflection will be done if a failure cannot otherwise be resolved.

We tested RILS on the same task as ROBBIE, introspecting about the domain-level reasoning processes, to verify that the cost of *case-based* introspective learning is not noticeably higher than that of the model-based reasoning ROBBIE uses. The costs of case retrieval turn out to be equivalent to the costs of manipulating the complex data structure of ROBBIE's model. In both cases, the cost of introspective learning is small enough to be dwarfed by the costs of producing output statements that report on the system's behavior. RILS, used in this way, shows the feasibility of a case-based introspective learner. Reusing CBR methods for all parts of the system allows a simple model to represent complex tasks.

Our preliminary tests of RILS also examined the additional overhead of one layer of reflection: if RILS introspectively analyzed its introspective processes but went no higher. The overhead appears significant, roughly three times as slow under ordinary failure-free processing conditions, and more slow during failure diagnosis. This underscores the importance of controlling reflection so that higher levels of abstraction are only considered when circumstances require it.

We have chosen for the time being to arbitrarily limit RILS to applying introspective reasoning only to its domain task, *unless a failure is detected which cannot be resolved or repaired by the introspective learner.* If a failure cannot be diagnosed or repaired, RILS will jump to a higher level of abstraction and introspect about its introspective process. This limits the overhead of reflection to a reasonable rarity. Work with RILS in this situation is currently incomplete. Future work will examine ways to incorporate more higher-level introspection with less cost, and ways to control reflection when un-diagnosed failures occur.

Conclusions

AI systems with meta-reasoning capabilities ought to interact better with complex domains or other agents than non-introspective systems. Meta-reasoning systems can respond more flexibly to new situations by altering their processes on the fly. Systems in complex domains must be adaptable; a human designer cannot build in a response for every eventuality. Introspective reasoning, in particular, has been shown to improve adaptability by permitting the system to learn to accommodate gaps in its original reasoning processes (Fox 1995; Stroulia 1994).

Non-reflective introspective reasoning can provide a significant benefit, but such systems fall prey to the criticism that they are merely rigid one level further back: they do not introspect reflectively.

RILS demonstrates that case-based reasoning can be used to manipulate an explicit model of reasoning processes, by embedding the introspective knowledge in a collection of cases. RILS also shows, at a preliminary stage, that reflective introspective learning can be done without sacrificing the existence of an explicit model of reasoning. In RILS the model is a collection of cases,

but still represents declaratively the entire reasoning process at both domain and introspective levels.

Our work on RILS is still preliminary. We can see already that the costs of reflection make it a dangerous tool to use. It seems clear that the only real solution at this point is to reflect only on an "as-needed" basis: RILS will only reflect to a higher level when it cannot solve the problem with a lower level of analysis. Future work will examine the issue of controlling reflection in more detail by comparing different control methods and their resulting costs.

References

Arcos, J., and Plaza, E. 1993. A reflective architecture for integrated memory-based learning and reasoning. In Wess, S.; Altoff, K.; and Richter, M., eds., *Topics in Case-Based Reasoning.* Kaiserslautern, Germany: Springer-Verlag.

Birnbaum, L.; Collins, G.; Brand, M.; Freed, M.; Krulwich, B.; and Pryor, L. 1991. A model-based approach to the construction of adaptive case-based planning systems. In Bareiss, R., ed., *Proceedings of the Case-Based Reasoning Workshop*, 215–224. San Mateo: DARPA.

Bradzil, P., and Konolige, K., eds. 1990. *Machine Learning, Meta-Reasoning and Logics.* Boston: Kluwer Academic Publishers.

Clancy, W. 1988. Acquiring, representing, and evaluating a competence model of diagnostic strategy. In Chi, M.; Glaser, R.; and Farr, M., eds., *The Nature of Expertise.* Hillsdale, NJ: Lawrence Erlbaum Associates. chapter 12, 343–418.

Collins, G.; Birnbaum, L.; Krulwich, B.; and Freed, M. 1993. The role of self-models in learning to plan. In *Foundations of Knowledge Acquisition: Machine Learning.* Kluwer Academic Publishers. 83–116.

Cox, M., and Freed, M. 1995. Using knowledge of cognitive behavior to learn from failure. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms.*

Cox, M. 1996. *Introspective multistrategy learning: Constructing a learning strategy under reasoning failure.* Ph.D. Dissertation, College of Computing, Georgia Institute of Technology. Technical Report GIT-CC-96-06.

Davis, R. 1982. Application of meta level knowledge to the construction maintenance and use of large knowledge bases. In Davis, R., and Lenat, D., eds., *Knowledge-Based Systems in Artificial Intelligence.* McGraw-Hill. 227–484.

Fox, S., and Leake, D. 1995. Modeling case-based planning for repairing reasoning failures. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms.* Stanford, CA: AAAI.

Fox, S. 1995. *Introspective Reasoning for Case-Based Planning.* Ph.D. Dissertation, Indiana University, Computer Science Department. IUCS: Technical Report 462.

Freed, M., and Collins, G. 1994. Adapting routines to improve task coordination. In *Proceedings of the 1994 Conference on AI Planning Systems*, 255–259.

Gurer, D.; des Jardins, M.; and Schlager, M. 1995. Representing a student's learning states and transitions. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms.* Stanford, CA: AAAI.

Ibrahim, M. 1992. Reflection in object-oriented programming. *International Journal on Artificial Intelligence Tools* 1(1):117–136.

Krulwich, B.; Birnbaum, L.; and Collins, G. 1992. Learning several lessons from one experience. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 242–247. Bloomington, IN: Cognitive Science Society.

Oehlmann, R.; Edwards, P.; and Sleeman, D. 1994. Changing the viewpoint: Re-indexing by introspective questioning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 675–680. Lawrence Erlbaum Associates.

Ram, A., and Cox, M. 1994. Introspective reasoning using meta-explanations for multistrategy learning. In Michalski, R., and Tecuci, G., eds., *Machine Learning: A multistrategy approach Vol. IV.* Morgan Kaufmann. 349–377.

Rosenbloom, P.; Laird, J.; and Newell, A. 1993a. *Meta Levels in Soar*, volume I. The MIT Press. chapter 26.

Rosenbloom, P.; Laird, J.; and Newell, A. 1993b. *R1-SOAR: An Experiment in Knowledge-Intensive Programming in a Problem Solving Architecture*, volume I. The MIT Press. chapter 9.

Smith, B. 1982. *Reflection and Semantics in a Procedural Language.* Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA. MIT-LCS-TR-272.

Stroulia, E., and Goel, A. 1995. Functional representation and reasoning in reflective systems. *Applied Artificial Intelligence: An International Journal, Special Issue on Functional Reasoning* 9(1):101–124.

Stroulia, E. 1994. *Failure-Driven Learning as Model-Based Self-Redesign.* Ph.D. Dissertation, College of Computing, Georgia Institute of Technology. Technical Report GIT-CC-96106.