

# Operational Rationality through Compilation of Anytime Algorithms

*Shlomo Zilberstein*

**H**ow can an artificial agent<sup>1</sup> react to a situation after performing the correct amount of thinking? My Ph.D. dissertation (Zilberstein 1993)<sup>2</sup> presents a theoretical framework and a programming paradigm that provide an answer to this question. The solution is based on the replacement of standard modules of a program with more flexible computation elements that are called *anytime algorithms* (Dean and Boddy 1988; Horvitz 1989). In addition, the solution includes an offline compilation process and a run-time monitoring component that guarantee that the agent is performing the correct amount of thinking in a well-defined sense.

## Control of Deliberation Time

Artificial agents must perform some real-time deliberation to solve such problems as path planning and task scheduling. An important aspect of intelligent behavior is the capability of agents to factor the cost of deliberation into the deliberation process. Two factors determine the cost of deliberation: (1) the resources consumed by the process, primarily computation time, and (2) constant change in the environment that might decrease the relevance of the outcome and, hence, reduce its value. The performance of artificial agents can greatly be improved by optimizing the quality of their decisions net of deliberation costs. The problem of deliberation cost has been discussed widely in philosophy, economics, and AI. In AI, researchers have proposed a

number of metalevel architectures to control the cost of base-level reasoning. The model that I developed for this dissertation belongs to this class of solutions. In particular, its metalevel reasoning component optimizes resource allocation to the base-level performance components. The resulting agent is said to be an operationally rational agent.

Operational rationality separates two, central aspects of agent construction: (1) the development of the performance components and (2) the optimization of performance. It introduces a new type of modularity into real-time-system construction. This

---

## *How can an artificial agent react to a situation after performing the correct amount of thinking?*

---

type of modularity is achieved by using anytime algorithms as the elementary components of the system.

## Anytime Algorithms

Anytime algorithms expand on the traditional view of a computational procedure because they offer to fulfill an entire spectrum of input-output specifications over the full range of run times. The quality of the results of such algorithms improves gradual-

ly as computation time increases; hence, anytime algorithms offer a trade-off between resource consumption and output quality. This trade-off is characterized by the performance profile of each algorithm. When combined with an appropriate metalevel control, anytime algorithms provide a powerful tool in constructing real-time agents.

A number of successful applications have been developed in such areas as path planning (Boddy and Dean 1989) and medical diagnosis (Horvitz 1989). The fundamental problem addressed in this dissertation is the extension of the property of gradual improvement from the level of single anytime algorithms to the level of complex systems that are composed of multiple anytime algorithms.

## Composing Anytime Algorithms

Modularity is widely recognized as an important issue in system design and implementation. However, the use of anytime algorithms as the components of a modular system presents a special type of scheduling problem (Russell and Zilberstein 1991).

In standard algorithms, the fixed quality of the output allows for composition to be implemented by a simple call-return mechanism. However, when algorithms have resource allocation as a degree of freedom, the question arises of how to construct, for example, the optimal composition of two anytime algorithms, one of which feeds its output to the other. This problem is solved by an offline compilation process and a run-time monitoring component that together generate a performance-maximizing profile for the complete system.

## Compilation

Given a system composed of anytime algorithms, the compilation process is designed to determine the optimal allocation of time to the components for any given total allocation. The crucial metalevel knowledge is kept in the anytime library in the form of conditional performance profiles. These profiles, an extension of an earlier notion of performance

description, characterize the performance of each elementary anytime algorithm as a function of run time and input quality.

My dissertation studies the complexity of compilation of various programming structures. In particular, the compilation of functional composition is shown to be NP complete in the strong sense. However, local compilation techniques, whose complexity is linear in the size of the program, are shown to be both efficient and optimal for a large class of programs. In addition, a number of approximate time-allocation algorithms are shown to solve the compilation problem efficiently in the general case.

The compilation process produces contract algorithms that require the determination of the total run time when activated. Some real-time domains, however, require interruptible algorithms whose total run time is unknown in advance. This problem is solved by a general method by which an interruptible algorithm can be constructed once a contract algorithm is compiled.

## Run-Time Monitoring

Monitoring plays a central role in anytime computation because it complements anytime algorithms with a mechanism that determines their run time. The dissertation examines the monitoring problem in two types of domain. One type is characterized by the predictability of utility change over time. High predictability of utility allows an efficient use of contract algorithms modified by various strategies for contract adjustment. The second type of domain is characterized by rapid change and a high level of uncertainty. In such domains, active monitoring is essential to guarantee operational rationality. Active monitoring schedules interruptible algorithms based on the value of computation criteria.

## Applications

The advantages of compilation and monitoring of anytime algorithms have been demonstrated through a number of applications. Zilberstein

and Russell (1993) describe one application in the area of mobile robot navigation. The robot is situated in a simulated environment with random obstacles. Initially, the robot does not have a map of the environment, but it has a vision capability that allows it to create an approximate map. The accuracy of the domain description depends on the time allocated to the vision module. Path planning is performed by an anytime search algorithm that varies the abstraction level of the domain description. The use of abstraction allows the algorithm to quickly find a low-quality plan and then repeatedly refine it by replanning a segment of the plan in more detail. Finally, the compiled performance profile of the system and the time-dependent utility function of the robot are used to monitor the execution of the sensing and planning modules and maximize the overall performance.

Preliminary results based on several applications are encouraging. They show that the model of operational rationality offers both a methodological and a practical contribution to the field of real-time decision making. The model simplifies the design of real-time deliberative systems and provides an efficient approach to resource-bounded reasoning. In collaboration with the Robotic Lab at the University of Massachusetts, I am currently applying the model to larger applications.

## References

- Boddy, M., and Dean, T. L. 1989. Solving Time-Dependent Planning Problems. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 979-984. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.
- Dean, T. L., and Boddy, M. 1988. An Analysis of Time-Dependent Planning. In Proceedings of the Seventh National Conference on Artificial Intelligence, 49-54. Menlo Park, Calif.: American Association for Artificial Intelligence.
- Horvitz, E. J. 1989. Reasoning about Beliefs and Actions under Computational Resource Constraints. In *Uncertainty in Artificial Intelligence, Volume 3*, eds. L. N. Kanal, T. S. Levitt, and J. F. Lemmer, 301-324. Amsterdam: North-Holland.
- Russell, S. J., and Zilberstein, S. 1991.

Composing Real-Time Systems. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 212-217. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Zilberstein, S. 1993. Operational Rationality through Compilation of Anytime Algorithms. Ph.D. diss., Computer Science Division, University of California at Berkeley.

Zilberstein, S., and Russell, S. J. 1993. Anytime Sensing, Planning, and Action: A Practical Model for Robot Control. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, 1402-1407. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

## Acknowledgments

I am deeply grateful to my adviser, Stuart Russell, for making my studies challenging and enjoyable. I am also thankful to the members of my dissertation committee—Alice Agogino, Tom Dean, and Susan Graham—and to the members of RUGS. This research was supported in part by the National Science Foundation and in part by the Malcolm R. Stacey Fellowship.

## Notes

1. An agent can be thought of as a robot situated in a particular environment and capable of translating perceptual input into actions that bring about a desired state.
2. A copy of this dissertation is available as technical report UCB/CSD-93/743 from the Computer Science Division, University of California, Berkeley, CA 94720. It is also available online through MOSAIC at GOPHER server. It is also available online through the author's HTTP server at <http://anytime.cs.umass.edu/~shlomo>.

Shlomo Zilberstein is an assistant professor of computer science at the University of Massachusetts at Amherst. He received his B.A. in computer science, summa cum laude, from the Technion, Israel Institute of Technology, in 1981, and his Ph.D. in computer science from the University of California at Berkeley in 1993. Zilberstein's current research interests include resource-bounded reasoning, autonomous agent architectures, real-time planning and monitoring, and decision-theoretic control of reasoning.