THE INSTRUCTIBLE PRODUCTION SYSTEM: A RETROSPECTIVE ANALYSIS

Michael D. Rychener
Carnegie-Mellon University

ABSTRACT

In building systems that acquire knowledge from tutorial instruction, progress depends on determining certain functional requirements and ways for them to be met. The Instructible Production System (IPS) project has explored learning by building a series of experimental systems. These systems can be viewed as being designed to explore the satisfaction of some of the requirements, both by basic production system mechanisms and by features explicitly programmed as rules. The explorations have brought out the importance of considering in advance (as part of the kernel design) certain functional components rather than having them be filled in by instruction. The need for the following functional components has been recognized:

- interaction language
- organization of procedural elements
- explanation of system behavior
- accommodation to new knowledge
- connection of goals with system capabilities
- reformulation (mapping) of knowledge
- evaluation of behavior
- compilation to achieve efficiency and automaticity

Since the experimental systems have varied in their effectiveness, some general conclusions can be drawn about the relative merits of various approaches. Seven such approaches are discussed here, with particular attention to the three whose behavior can be most effectively compared, and which reflect the temporal development of the project.
CHAPTER 14: THE INSTRUCTIBLE PRODUCTION SYSTEM

14.1 THE INSTRUCTIBLE PRODUCTION SYSTEM PROJECT

The Instructible Production System (IPS) project [Rychener & Newell, 1978] was begun in the fall of 1975 to study the construction and behavior of large-scale systems of production rules. Our hypothesis, extrapolated from work in cognitive psychology [Newell & Simon, 1972], was that intelligence would result, as a system grew in size, from an ability to deal with more situations and to apply more knowledge to solve problems. The motivation to use production systems had the same source [Newell & Simon, 1972]. To increase the scientific interest of building such systems, and ultimately to improve the chances of continuing growth and viability, it was stipulated from the start that the system was to be built by gradual “instruction” rather than by deliberate programming.¹

The research evolved into a series of explorations of the design of a starting system (KERNEL), from which the much larger system would be grown. The explorations spanned a four-year time period, until mid-1979, and involved the efforts of over a half dozen people.²

The setting in which instructional experiments took place was chosen to be one of “learning by doing”. In this paradigm, the instructor of the system watches and advises the system while it is solving problems in its chosen domain of expertise (see the work of Anzai and Simon [1979]). This is a good way to study learning because it combines attributes of both learning by being told and learning by independent exploration, while avoiding some of their drawbacks. That is, the instructor still instructs by telling, but the fact that the system is doing something at the same time allows the instructor to verify (partially) that the new knowledge is appropriate to the system’s current knowledge. In addition, the system is in a sense exploring in an environment that has new situations for it, under the guidance of the instructor and in the framework of problems posed by the instructor. When new knowledge interacts in some way with the system’s existing knowledge, that interaction has the greatest chance of being understood in the context of a situation where that knowledge is being applied. The system is forced to deal with new situations in its own way, using its own conceptual system, with the extra help of the instructor’s advice. But advice to the system is often limited, in that the system’s knowledge may not be stored so as to be brought to bear in all appropriate situations, and in that the

¹Actually, production systems are quite difficult to program, so an instruction mode has the potential of bringing a large system into the realm of feasibility. What is desired is that the production system itself be able to manage its knowledge, find interactions of new knowledge with old [Rychener, 1975], check consistency, formulate and select answers for questions that arise when new and old knowledge statements are compared, and do assorted other tasks that can’t even be predicted at this time. To complete this knowledge management task would require a great deal of knowledge itself, and the IPS project has only begun to realize what might be required for this much larger research goal.

²See the Acknowledgments near the end of the paper.