Input/output automata provide a formal model, applicable in describing most types of an asynchronous concurrent system. On its own, the I/O automaton model contains a very basic structure that enables it to model various types of distributed systems. To describe specific types of asynchronous systems, additional structure must be added to this basic model. The model presents an explicit method for describing and reasoning about system components such as processes and message channels that interact with one another, operating at arbitrary relative speeds.[1] The I/O automata were first introduced by Nancy A. Lynch and Mark R. Tuttle in "Hierarchical correctness proofs for distributed algorithms", 1987.[2]

"An I/O automaton models a *distributed system* component that can interact with other system components. It is a simple type of state machine in which the transitions are associated with named *actions*."[1] There are three types of actions: *input*, *output*, and *internal* actions. The automaton uses its input and output actions to communicate with its environment, whereas the internal actions are only visible to the automaton itself. Unlike internal and output actions that are selected and carried out by the automaton, the input actions – which simply arrive from the environment - are not under automaton's control.[1]