Job Life Cycle Model

The life cycle of an SPK job is represented as a finite state automaton.

Table of Contents

Introduction

A job is created when the user combines a model with data and submits this package to be run by Spk. Submitting jobs is one of the functions of the Model Design Agent (MDA), the graphical interface that runs on the user's workstation. The job is sent via the Internet to the Application Server (ASPK) to be translated into computer source code. Once compiled, it is sent onward, via the Internet to the Computational Server (CSPK) to be translated into machine code and executed.

The life cycle of a job can be represented as a sequence of states. When a user inquires about the status of one of her outstanding jobs, the answer is based on its current state. The history of a job is the sequences of states it has assumed, along with the times of transition from one state to another and any outputs that have occurred. The purpose of this document is to define the set of states that a job may pass through, and the rules governing transitions from one state to the next.

Job State Diagram

The following diagram represents the life cycle of a job as a finite state automaton. Four kinds of elements are present:

- States are represented by boxes.
- Transitions are represented by arrows.
- An event is represented by text appearing near the beginning of the arrow representing the transition caused by the event.
- An output is represented by text appearing near the end of a transition arrow. Not every transition produces output. To further distinguish an output from an event, the name of the output is preceded by the '>' character.

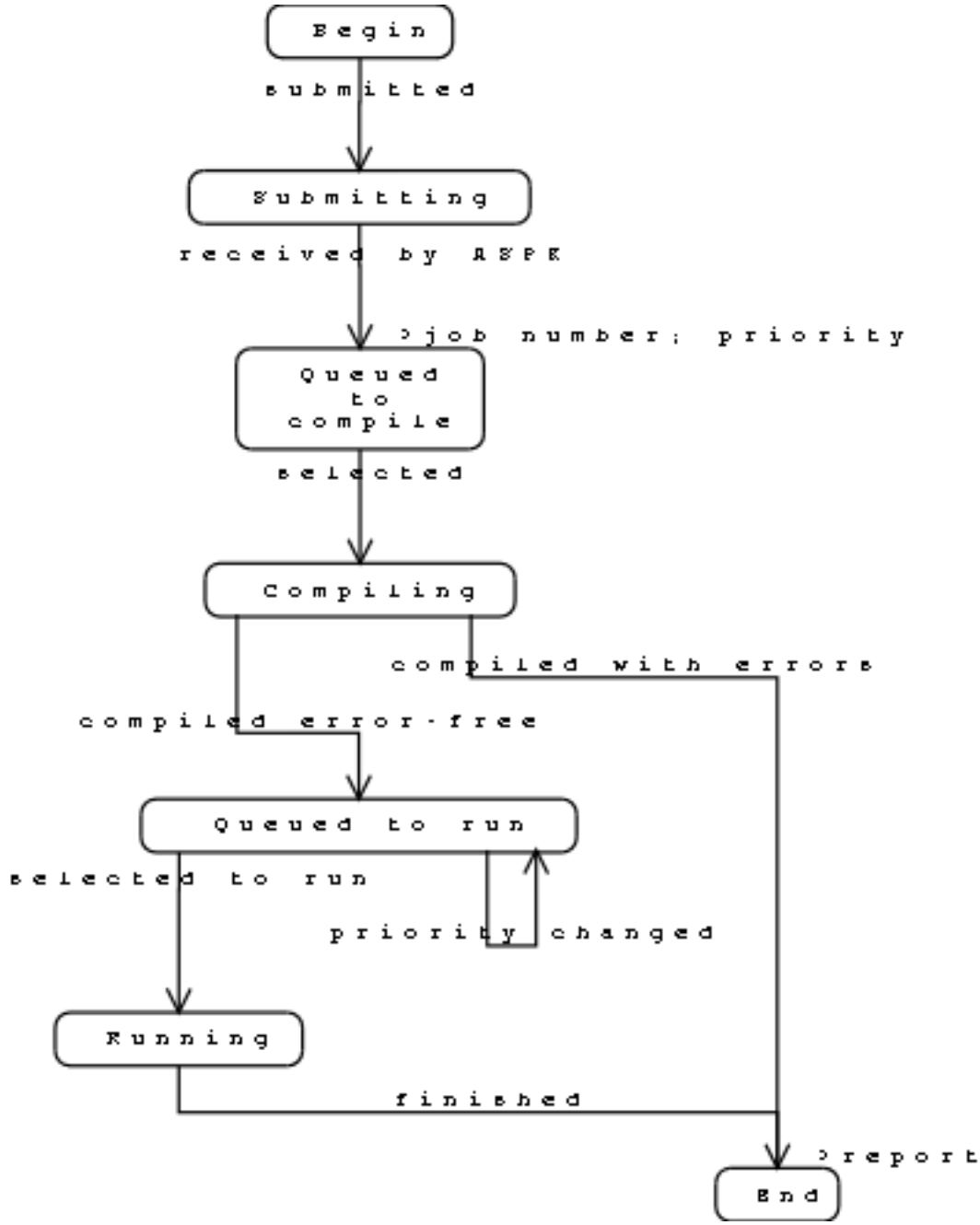

Figure 1. Job State Diagram

Description of Elements

Begin

This state is purely formal. The job is in this state until it is submitted.

submitted

This event occurs when the user commands the MDA to submit the job. At that time, the job enters the *Submitting* state.

Submitting

In this state, the job is in a sort of "limbo", which persists until it has been received by the ASPK.

received by ASPK

This event occurs when the job has been received by the ASPK. It then makes the transition to the *Queued to compile* state.

The transition produces *output*: a job number and a priority. The job number is unique for this Spk installation for all time.

Queued to compile

The job is waiting in a prioritized queue until sufficient resources become available for the compiler to translate it from model specifications to computer source code.

selected

occurs when compilation resources are available, there is no job of higher priority and there is no job of equal priority that has been waiting longer.

Compiling

The compiler is processing the model specification.

compiled error-free

When this event occurs, the job is ready to be queued for running

compiled with errors

If there are errors in the compilation, the job cannot be queued to run. Instead, the next state is *End*.

An error report is output on this transition.

Queued to run

The job is waiting in a prioritized queue for resources to assigned to it so that it can run.

selected to run

The job is selected when sufficient resources become available for it to run, there is no job of higher priority in the queue, and there is no job of equal priority that has been waiting longer.

priority changed

The priority of the job is changed. It makes the transition back to the same state, but with the new priority.

Running

The Computational Server (CSPK) is executing the job.

finished

When finished (whether the problem converges satisfactorily or not) the job comes to its *End* state.

A report is output on this final transition.

End

This is the final state of every job.