Often integrating ordinary differential equations or differential algebraic equations (DAE) do not constitute the problem alone. A common complement is finding the root of an algebraic function (an event function) that depends on the states of the problem. This formulation of a model enables the possibility of including discontinuities, an important part of the Functional Mock-up Interface (FMI) standard which allows hybrid models of differential algebraic equations. The problem of root-finding during integration is however difficult, both in a theoretical aspect and as a software problem.

The Illinois algorithm was chosen as a basis for the event algorithm. An important improvement is to apply the domain formulation, meaning that instead of defining an event as a change in sign (zero-crossing formulation) for the event function, \( g \), it is defined as a change in domain from \( g > 0 \) to \( g \leq 0 \) or vice versa, this is consistent with the FMI standard. An advantage is that the zero is no longer a special case and also, more importantly, events caused by event functions becoming exactly zero for a finite time is found correctly.

An implementation of software for the event algorithm is done in Assimulo, a Python/Cython wrapper for integrators. This enables event location for numerous integrators and therefore also support for simulating FMUs by using PyFMI.