

What is and what is not Event Processing?

Mani Chandy, California Institute of Technology

The general area of Event Processing (EP) will become a discipline only when we, the community of engineers, agree on (a) what it is, and equally importantly (b) what it is not. Almost all systems process events. If EP is the area that deals with analyzing and designing systems that process events then EP covers everything. An area that covers everything is an area that covers nothing. We, the community of people interested in event processing, here at Dagstuhl, must come to an understanding – however tentative and however informal – about what EP is and what it is not.

A discipline is characterized by bodies of theory, practice, and a class of application domains. Consider, for example, a broad discipline such as Operations Research. An OR department teaches – at its core – disciplines and applications based on probability and statistics, optimization and simulation. OR departments may teach much more; however, we have expectations that a person who has specialized in OR knows, for example, how to apply probability theory to practical problems. Likewise, we would not expect that all OR specialists know areas such as VLSI design. The discipline of OR has a rough, fuzzy boundary characterizing what it includes and what it does not. What is the boundary for EP? This is the fundamental question for all of us here at Dagstuhl.

One of the goals of this Dagstuhl workshop is to help develop a community of practitioners and researchers in EP with regular conferences, a Special Interest Group (SIG), journals and so forth. An EP conference should include sessions on most – if not all – the areas that are inside the EP boundary. Areas outside the EP boundary may or may not be included. Therefore, agreeing on the boundary is not merely a theoretical exercise: it has significant practical consequences.

Let me suggest some ideas that can help serve as a strawman proposal for what EP is. The only function of the strawman is as a vehicle to get a conversation going.

EP includes 5 components:

1. Sensors
2. Actuators
3. Compute engines
4. Information repositories
5. System managers

Many types of systems have these 5 components so we need to refine this further. EP deals with abstractions of sensors and actuators and not necessarily with their actual construction. Likewise, EP compute engines deal specifically with processing responses to events, i.e., state changes. Similarly, EP information repositories deal with storage of event information, and EP system managers deal with the management of the other 4 components.

EP is based on many other disciplines; for example, the compute engines may include simulators and other predictive models from OR and control theory. What makes EP different is its emphasis on responding to significant state changes.

Many factors are considered in the design of event-based systems. These parameters include the rates at which messages are generated by sensors; the network topology by which information about events is communicated; whether information is propagated continuously by analog signals, periodic synchronous messages, asynchronous messages or mobile agents; the computation carried out at different points in the network; and accuracy of sensors, actuators and computational nodes. We at Dagstuhl should develop mathematical models that aid systematic design by providing frameworks for evaluating design alternatives.

EP can develop into a discipline in the next decade with a collection of theories, design patterns, applications, and many industrial products. Given the rapid changes taking place, EP could be one of the most *useful* disciplines as well as being intellectually challenging.