Chapter 19
Panel:
Current State and Future of Event-Based Systems

Annika Hinze
University of Waikato, New Zealand

Jean Bacon
University of Cambridge, UK

Alejandro Buchmann
Technische Universität Darmstadt, Germany

Sharma Chakravarthy
The University of Texas at Arlington, USA

Mani Chandi
California Institute of Technology, USA

Avigdor Gal
Israel Institute of Technology, Israel

Dieter Gawlick
Oracle Incorporated, USA

Richard Tibbetts
StreamBase Systems, USA

ABSTRACT
This chapter is a panel discussion in writing. The field of event-based systems finds researchers from a number of different backgrounds: distributed systems, streaming data, databases, middleware, and sensor networks. One of the consequences is that everyone comes to the field with a slightly different mindset and different expectations and goals. In this chapter, we try to capture some of the voices that are influential in our field. Seven panellists from academia and industry were invited to answer and discuss questions about event-based systems. The questions were distributed via email, to which each participant replied their initial set of answers. In a second round every panelist was given the opportunity to expand their statement and discuss the contributions of the other panellists. The questions asked can be grouped into two types. Questions in the first group refer to each participant's understanding of the basic concepts of event-based systems (EBS), the pros and cons of EBS, typical assumptions of the field and how they understood EBS to fit into the overall landscape of software architectures. The second group of questions pointed to the future of EBS, possible killer applications and the challenges that EBS researchers in academia and industry need to address in the medium and long term. The next section gives each panellist's initial statements as well as their comments to other participants' contributions. Each participant's section starts with a short introduction of the panellist and their work. In the final section, we compare and reflect on the statements and discussions that are presented by the seven panellists.

DOI: 10.4018/978-1-60566-697-6.ch019
Panel: Current State and Future of Event-Based Systems

PANELLISTS’ CONTRIBUTIONS

Jean Bacon

Jean Bacon is Professor of Distributed Systems at the University of Cambridge Computer Laboratory, UK. She leads the Opera research group, with focus on middleware for large-scale, widely distributed or geographically concentrated ubiquitous systems. The Opera group pioneered event-based systems from the early 1990s, with applications including healthcare, pollution and transport monitoring, typically comprising multiple administration domains. Opera has carried out substantial work on access control, security and privacy for these and other applications, and the specification and enforcement of policy.

Jean is a Fellow of the IEEE and BCS and was an IEEE Fellowship Committee member in 2008. She was a member of the Governing Body of IEEE-CS, 2001-2007; and founding EIC of Distributed Systems Online 2000-2008, IEEE’s first online-only magazine. She is currently on the Editorial Board of Computer, IEEE-CS’s flagship magazine.

Annika: Where do you see the difference between publish/subscribe, event-based systems, Complex event processing (pub/sub, EBS, CEP)? How do you see message filtering technologies fitting into the picture? [Q1]

Jean: I suggest starting the discussion by giving some basic definitions:

Primitive event: Some define a primitive event as representing a state change (being notified asynchronously to interested parties). Others may accept the asynchronous transmission of a value e.g. a static sensor reading, as an event, even if there has not been a change of state in the process being monitored by the sensor.

We might extend the definition to include an event’s manifestation as a message e.g. with topic/type, attributes and their values, plus a system-generated source timestamp that might be a point timestamp with some tolerance or an uncertainty interval to allow for the impossibility of representing time exactly.

One could regard the difference in time, if not in attribute values, as a state change for consistency with the definition, but this would be against the intuition for “tell me only when some specified attribute changes value (significantly - by some specified amount).”

If mobile publishers are supported then a location stamp as well as a timestamp may be included in a message, or location might be deemed an attribute of an event.

Composite event: An expression combining primitive and composite events using a number of defined operators. The representation of a composite event raises many issues, as does engineering detection in a distributed system.

“Event-based system” is a general term that includes many different system styles. Common to all is asynchronous communication between decoupled communicating entities. EBSs include GUIs and server architectures (but these are sufficiently well-known, centralised and homogeneous not to be covered in this book), sensor-based systems in ubiquitous computing, active database triggers, stream processing, complex event processing, publish/subscribe applications such as news/stockquote notifications.

In order to address the questions below we need to distinguish client and system components’ functionalities and interfaces and colocation. For example, do we assume a dedicated network of event brokers to implement communication or might there be a peer-to-peer model with each node in a system containing both application and system components?

I see EBS as the umbrella term for all event-driven systems.

The differences lie in the characteristics and requirements of the supported applications such as security, reliability of communication, whether a transactional model is needed by the client level,