What the history of event-based systems tells you about your PhD

Alessandro Margara
https://margara.faculty.polimi.it
PhD expectations
PhD expectations

“Solve real-world challenges”

“Have a long-lasting impact on science”

“Make groundbreaking discoveries”

“Produce revolutionary new ideas in the research field”

“Contribute to make the world a better place”
PhD reality

Bad news: sorry, that’s (probably) not gonna happen!

Good news: you do have the chance to contribute to the evolution of your research field
A (biased and incomplete) history of event-based systems
Research fields are in continuous (rapid!) evolution

Research is a collective effort: every contribution (and discussion) counts and steers the evolution of a field

Research is multi-disciplinary: cross fertilization across domains is inevitable and vital
Publish-subscribe

Publish-subscribe

Pietzuch et al. “Composite event detection as a generic middleware extension”. IEEE Network, 2004
Li, Jacobsen “Composite subscriptions in content-based publish/subscribe systems”. Middleware, 2005
Event processing systems

Producers

Event processing system

Consumers
Event processing systems
A language for event processing

define Fire(area = $a)
from Smoke(area = $a) and last
Temp(value>40, area = $a) within 5 min from Smoke
and not Rain (mm>2, area = $a) between Temp and Smoke

Consuming Temp

Cugola, Margara “TESLA: a formally defined event specification language”. DEBS, 2010 (Test of Time Award in DEBS 2020)
A language for event processing

• TESLA was formally defined using a metric temporal logic

• The topic of defining a standard language and semantics for event processing is still an open research question

• Dagstuhl seminar on the topic in 2020

We are not alone in the universe!

Stream processing in databases

- A stream is an append only (unbounded) table

- We can use the same (relational) processing abstraction for both tables and streams

- Queries on streams never terminate, but keep updating their solutions as new data enters the input streams

Stream processing in databases

• How to build a unifying abstraction for both static and dynamic (streaming) data?
  • How to integrate concepts like transactional semantics

• How to build a unifying system or software architecture to handle both static and dynamic data

Stream processing in programming languages

Reactive variables

A

B = f(A)

C = g(A)

D = h(B, C)

Margara, Salvaneschi “We Have a DREAM: Distributed Reactive Programming with Consistency Guarantees”. DEBS, 2014.
Stream processing and programming models


https://github.com/deib-polimi/noir
Research fields are in continuous (rapid!) evolution
In 15 years, from pub-sub to distributed stream-processing systems

Research is a collective effort: every contribution (and discussion) counts and steers the evolution of a field
Similar ideas emerged from different groups and communities, consolidating over the years

Research is multi-disciplinary: cross fertilization across domains is inevitable and vital
Recognizing stream processing problems in different areas enabled applying/adapting the same solutions
Conclusions

You are (probably) not gonna make a groundbreaking discovery …

... but you DO have the opportunity to bring your contribution and shape your area of research!

• It’s a collective effort
• You’ll see the results within only a few years
Suggestions

Aim at a deep and clear understanding of the area

As the state of things evolve rapidly, it is difficult to find detailed descriptions, reviews, models, classifications. They bring a value!

Focus on quality and precision

It may require more time, but in the long term it will pay off. Doing a PhD is a unique opportunity to dedicate all/most of your time to study a subject in depth.

Talk to other researchers and communities

Observing the same concepts from different perspectives expands your horizons.
Time for discussion!