

# Web services, choreography and the Event Calculus

A short introduction

# History

- Invented by Kowalski & Sergot (1986)
- Many variants
- Shanahan's tutorial paper
  - <http://casbah.ee.ic.ac.uk/~mpsha/pubs.html>
  - He uses it for robotics, reasoning about change
- Explored by Sloman for use in expressing policies

# Event Calculus

## concepts

- Events
  - Something happened
- Fluents
  - Something is true of the world
- Time
  - Partially ordered set of time points

# Event

- Occurs at a point in time
- May initiate the truth of a fluent
- May terminate a fluent
- May 'release' a fluent
  - (The truth of a released fluent is no longer known)

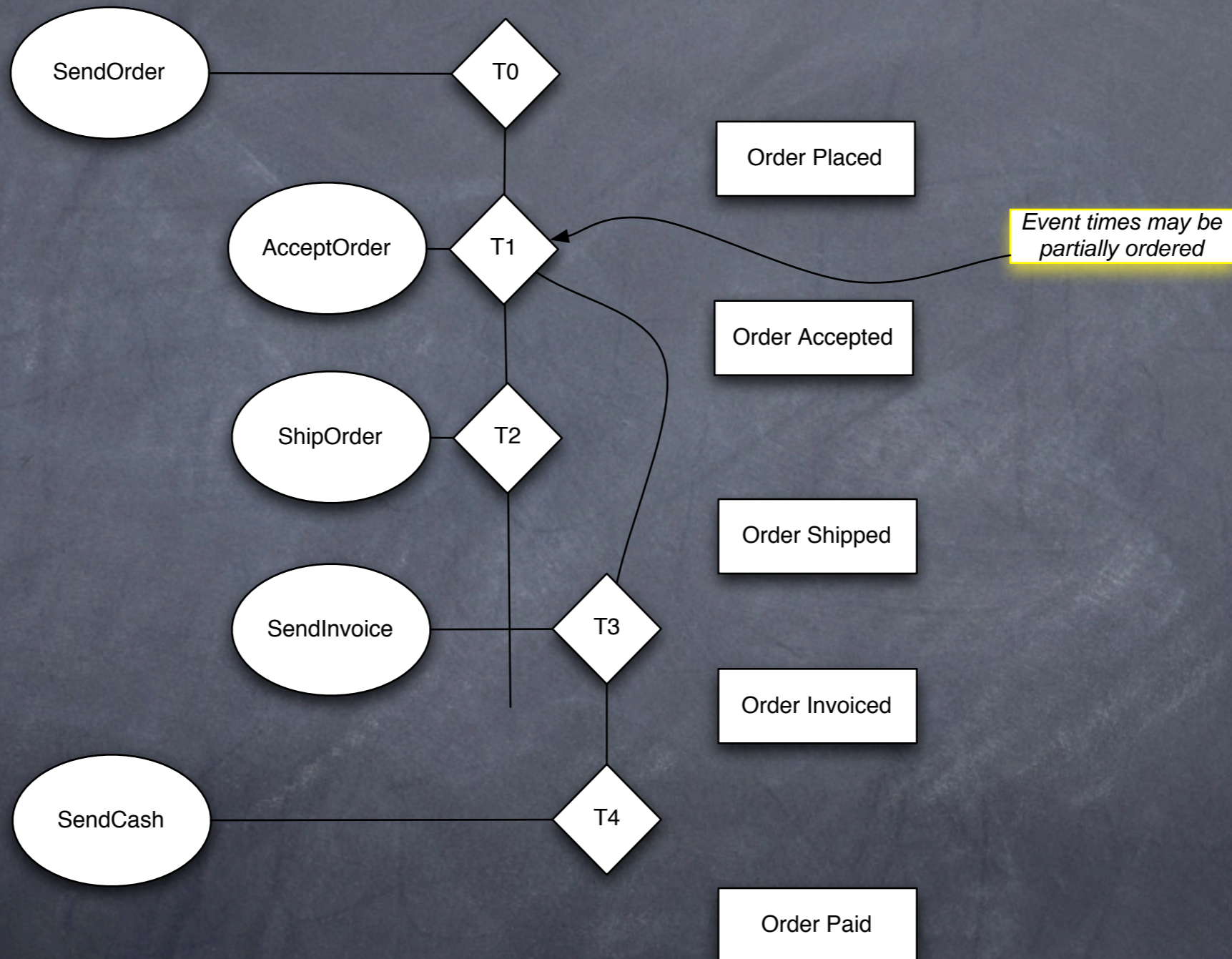
# Fluent

- A logical formula whose truth (or falsity) has an extent in time
- Fluents are given concrete names
  - I.e., are first class entities
- Fluents are initiated, terminated by events
- Fluents may hold at a particular time
- Fluents can be clipped, released

# EC Logic

- We can reason about what is true
- Uses standard predicate logic with a few extras - closed world assumption/  
circumscription

# An order



# Simple axioms of EC

Fluent

Time point

- $\text{HoldsAt}(f,t) \leftarrow \text{Initially}(f), \neg \text{Clipped}(0,f,t).$
- $\text{HoldsAt}(f,t) \leftarrow \text{Happens}(a,t_1), \text{Initiates}(a,f), t > t_1, \neg \text{Clipped}(t_1,f,t).$
- $\text{Clipped}(t_1,f,t_2) \leftarrow \text{Happens}(a,t), t_1 < t < t_2, \text{Terminates}(a,f)$



# Simple Order logic

- Happens(sendOrder, T0).
- Happens(acceptOrder, T1).
- Happens(sendCash, T4)
- Initiates(acceptOrder, orderAccepted).
- Initiates(sendCash, orderCompleted).

# Role in choreography

- Complements pi-calculus view
  - not particularly sensitive to process abstraction
- External state modeled as a set of fluents
- Transitions/messages are marked by events
- Permits expression of verification conditions
- Permits expression of policies