# Basic Concepts in Dependable Real-Time Systems

H.Kopetz

January 2012

# Outline

- Introduction
- Basic Concepts
  - Time
  - Component
  - Variable and Message
  - Behavior, Service--Failure
  - State--Error
- Determinism
- System of Systems
- Emergence
- Conclusion

# Introduction

- We understand the world by modeling
  —***models of reality*** .
- A model is a reduced representation of a scenario driven by a purpose.
- It is not possible to verify a model—it can only be falsified (Popper on the *coherence theory* of truth).
- Many basic concepts in computer science are derived from mathematics—***but mathematics is timeless!***

# Model Building

Model building is driven by

- Utility—to explain some basic or advance need
- Parsimony (Occam's razor)—simple and understandable
- Generality—applicable to many scenarios

But there are *inherent* conflicts!

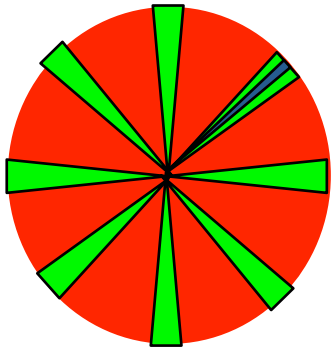
Whenever we speak about *basic concepts* we have to wrestle with the limits of language (Wittgenstein).

*The words fell apart in my mouth just as rotten wood.*
*Die Worte zerfielen mir im Munde wie* ***modrige Pilze***
Hugo v. Hofmannsthal, 1902

# What is *Time*?

- Time is an *external quantity* that changes according to known physical laws.
- Our model of time is based on *Newtonian* physics
- The flow of time is *unidirectional*, from the past to the future. An  a cut of the time-line is called an *instant*.
- With proper instruments (clocks), the flow of time can be observed and measured with a *given limited accuracy*, but it cannot be controlled.
- The astronomical second is *not constant*: duration of the tropical year (Withrow 1988, p.187):
  - 45 BC  365.24232 days
  - 1900 AD  365.24222 days
- True simultaneity in the physical world cannot be measured.

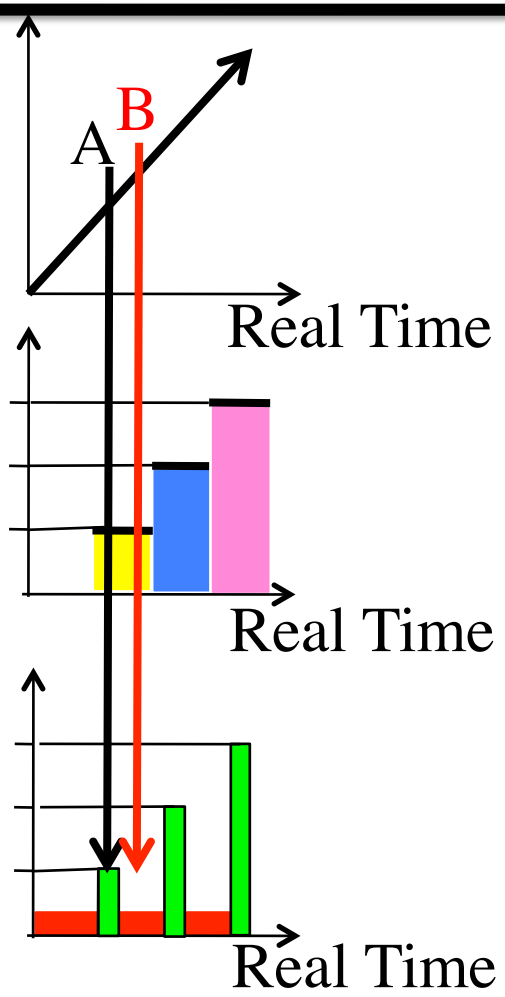
# Models of Time in Cyberphysical Systems (CPS)

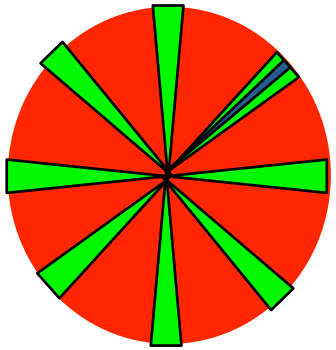
**Dense**
Physics

**Discrete**
Central Computer

**Sparse**
Distributed
Computer



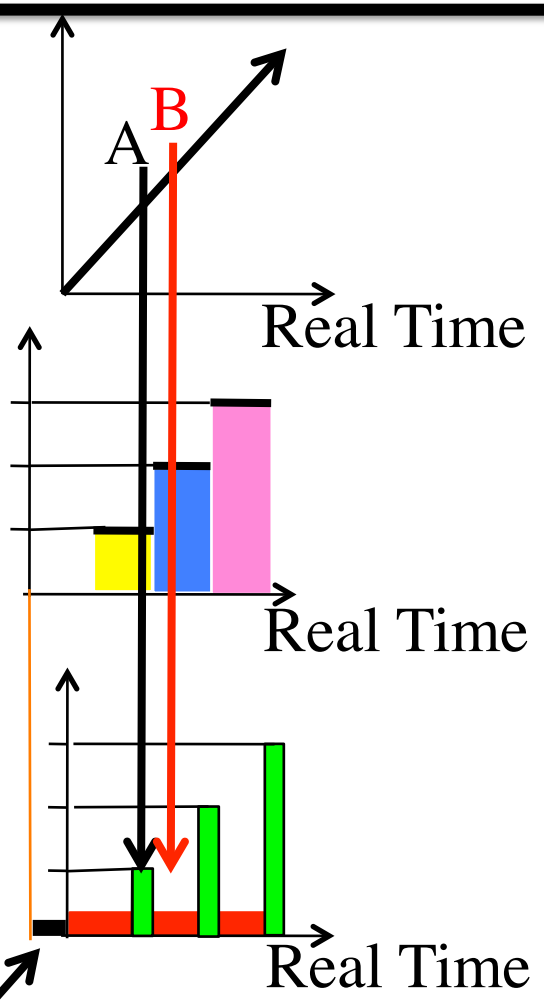
# Models of Time in the CPS

**Dense**
Physics

**Discrete**
Central Computer

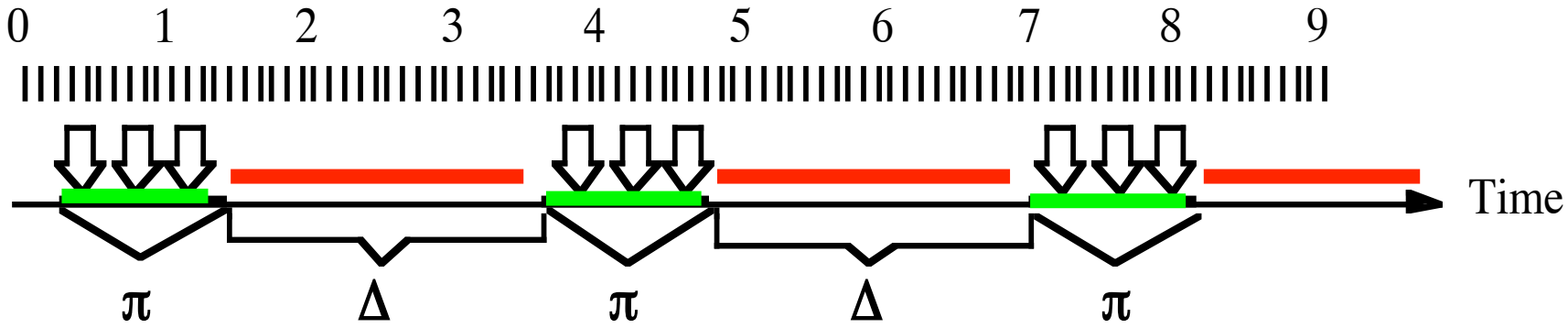
**Sparse**
Distributed
Computer



# *Sparse Time* Mode

Whenever we use the term *time* we mean *physical time* as defined by the international standard of time TAI.

If the occurrence of events is restricted to some active intervals on the timeline with duration $\pi$ with an interval of silence of duration $\Delta$ between any two active intervals, then we call the time base $\pi/\Delta$-sparse, or ***sparse*** for short, and events that occur during the active intervals ***sparse events***.




Events ⇩ are only allowed to occur at subintervals of the timeline

# What is a Component?

- ***Hardware/software unit*** that accepts input messages, provides a useful service, maintains internal state, and produces after some *elapsed time* output messages containing the results. It is aware of the progression of *physical time*
- ***Unit of abstraction***, the behavior of which is captured in a *high-level concept* that is used to capture the services of a subsystem.
- ***Fault-Containment-Unit (FCU)*** that maintains the abstraction in case of fault occurrence and contains the immediate effects of a fault (a fault can propagate from a faulty component to a component that has not been affected by the fault only by erroneous messages).
- ***Unit of restart, replication and reconfiguration*** in order to enable the implementation of robustness and fault-tolerance.

# Variable and Message

- A ***variable*** is a Triple
  - *Name*: denotes a *concept*
  - *Value:* denotes an attribute of the concept, the *data*
  - *Time:* denotes the time of observation
- A ***message*** is an information structure that is formed for the purpose of inter-system (or inter-component) interaction. A message covers the *data aspect* and *the temporal aspect* of an interaction.
  - The ***temporal aspect of an interaction*** is concerned with the instants of sending and receiving of a message.
  - The ***data aspect of a message*** relates to a set of variables.

# *R-(Real-time) State* of a Component--Error

*The R-state enables the determination of a future output solely on the basis of the future input and the state the system is in. In other word, the R-state enables a "decoupling" of the past from the present and future. The R-state embodies all past history of a system. Knowing the R-state "supplants" knowledge of the past. . . . Apparently, for this role to be meaningful, the* ***notion of past and future*** *must be relevant for the system considered*.

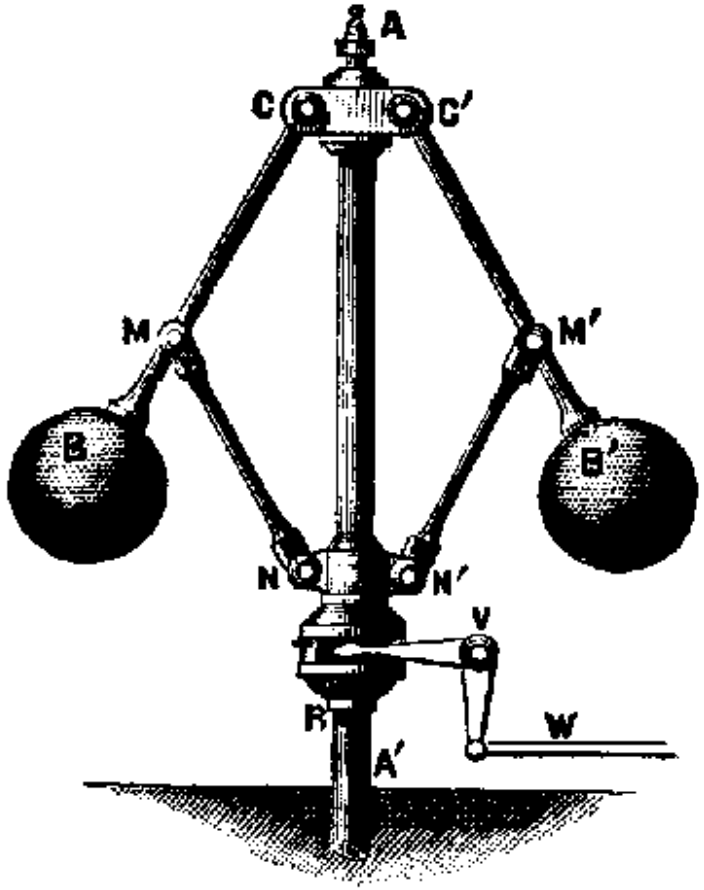
Mesarovic, p.45

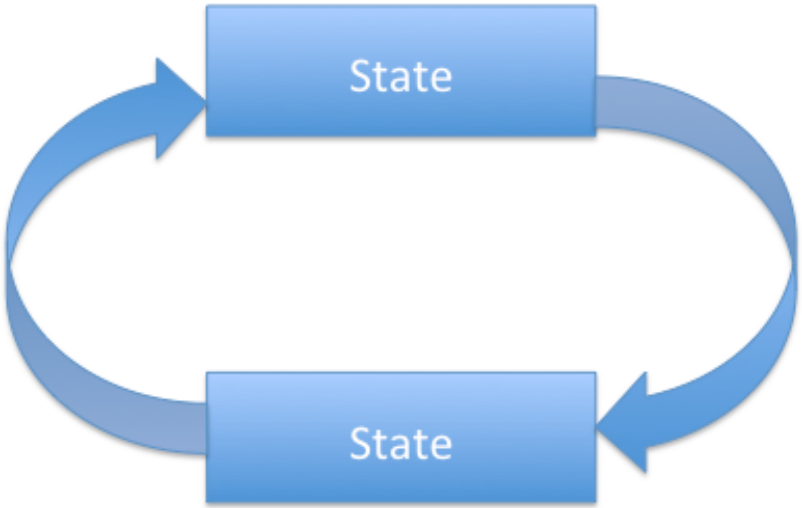
**Error**: Unintended state

# On the *Notion* of State

- *State* is a fundamental concept of computer science!
- Does the notion of *state, as used by the computer science community,* include real-time?
- When we talk about *state* in *state machines*, is at the *R-state* or is it a different concept of *state*, say *C (computation)-state*?
- In may opinion *C-state* and *R-state* are orthogonal concepts—we should be careful not to confuse.

# Causality versus Feedback

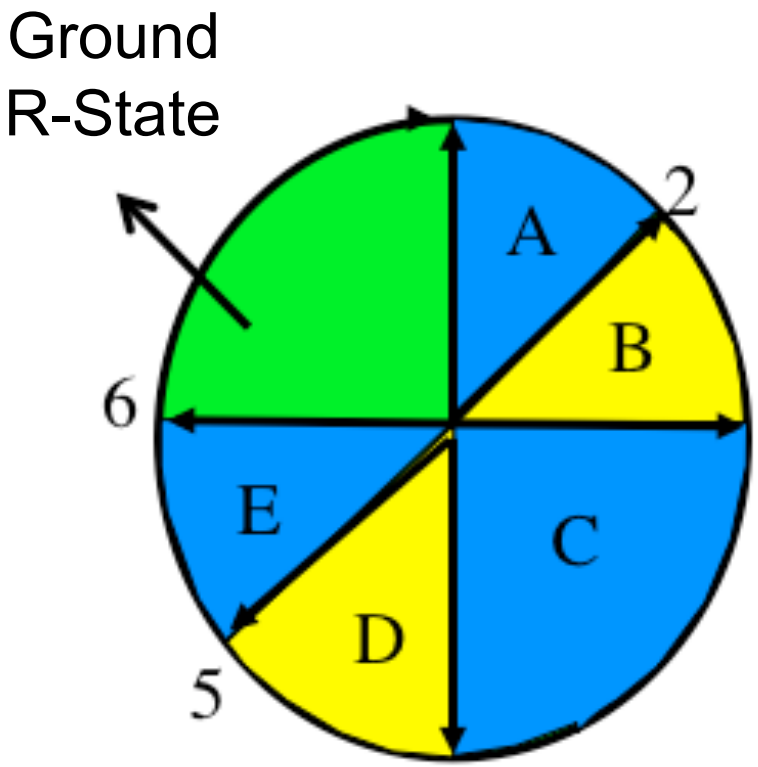


# Ground R-State

The ground R-state of a component is the R-state of a component at an instant when all tasks are inactive and where all communication channels are empty.

A relevant ground R-state must be provided to reintegrate a component after a hardware reset.



# *Behavior* and *Service* of a Component--*Failure*

- ***Behavior of a Component***: sequence of timed output messages (influenced by input and R-state).
- ***Service:*** Intended Behavior
- ***Failure***: Deviation of actual behavoir from intended behavior at an instant.

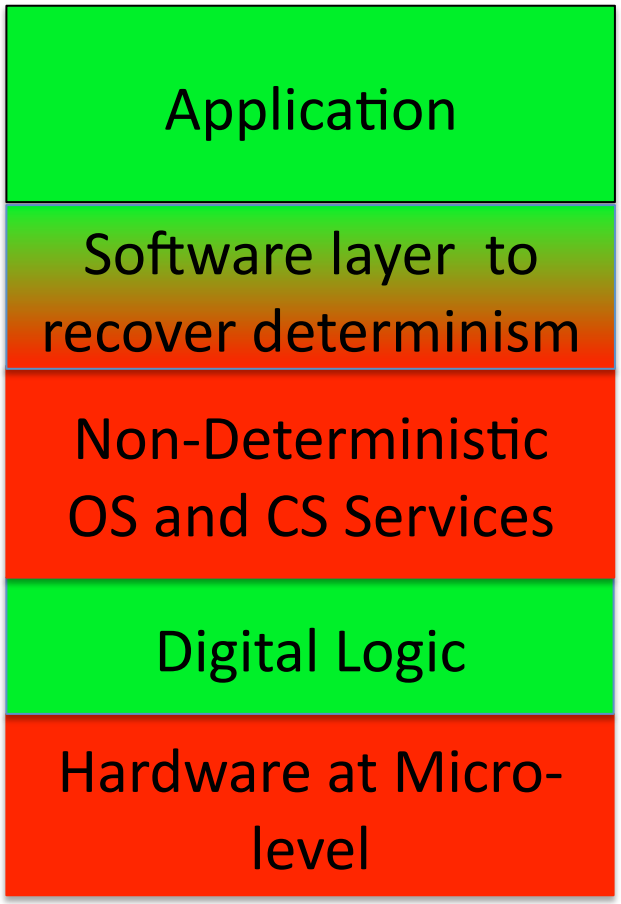
# Definition of Determinism

A physical system behaves deterministically if, given an initial state at instant t and a set of future timed inputs, then the future states and the ***values and instants*** of future outputs are entailed.

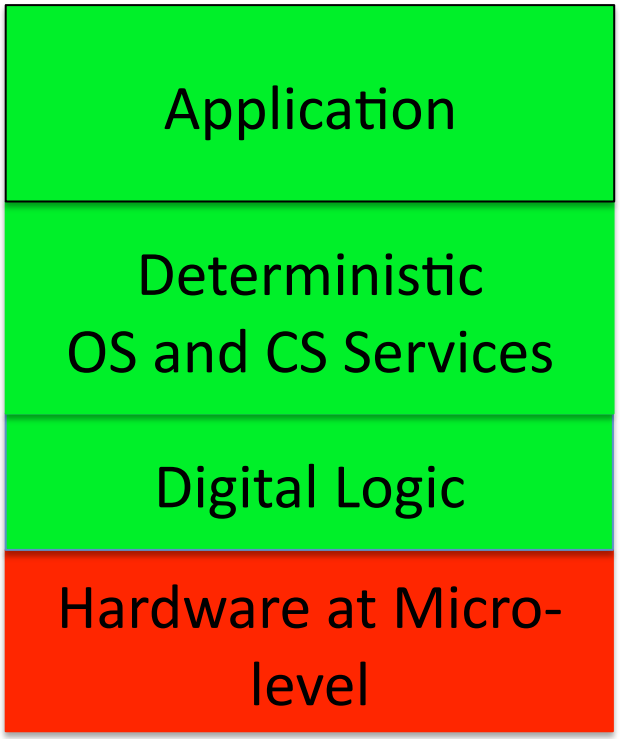
# On the Notion of *Determinism*

- Determinism is a property of a computation that makes it possible to predict the future behavior, given the present R-state and the timed inputs.
- The computer-science notion of determinism, which *eliminates the temporal dimension*, is not appropriate for real-time systems.
- In many real-time applications, determinism is required at the application level. Example: application of the brakes in a car.
- It is an interesting research issue how to recover determinism above non-determinate behavior.

# Recovery of Determinism




## TTA Approach



# Definition: System of Systems (SoS)

**We call a distributed system a *System of Systems (SoS)* if some actions that are needed to arrive at a solution are delegated from the system that interfaces with the *problem owner* to other *autonomous problem solving systems (also called* constituent systems*)*.**

- The autonomous *problem solving systems* (PSSs) that participate in an SoS can be under different management control, of heterogeneous technology, geographically distributed, conform to different architectural styles, and can operate on any scale from fully automatic to control by humans.
- The ***primary PSS*** does not have administrative-control over the secondary (called) PSS.
- A ***secondary PSS*** may reject a request from the primary PSS, based on the established incentive structure.

# Fundamental *Paradigm Shift* in SoS Design

| Characteristic | Old-Classic | New-SoS |
|---|---|---|
| Scope of System | Fixed (known) | Not known |
| Specification | Fixed | Changing |
| Control | Central | Distributed |
| Evolution | Version contr. | Uncoordinated |
| Testing | Test phases | Continuous |
| Technology | Given and fixed | Uncertain |
| System development | Process model | ??? |

# Definition of Emergence

A system exhibits *emergence* when there are coherent qualities at the macro-level (the ***abstract A level***) that dynamically arise from the interactions between the parts at the micro-level (the ***base level B***). Such qualities are novel w.r.t. the individual parts of the system (DeW05).

The definition above uses the concept of an 'quality' as a general term to denote the result of the process of emergence: *properties, behavior, structure, patterns*, etc.

# Emergent Property Requires a *New Concept*

In many instances, the emergent property is of a new kind, described by a new concept, that is foreign to the established theory that deals with the parts and their interactions.

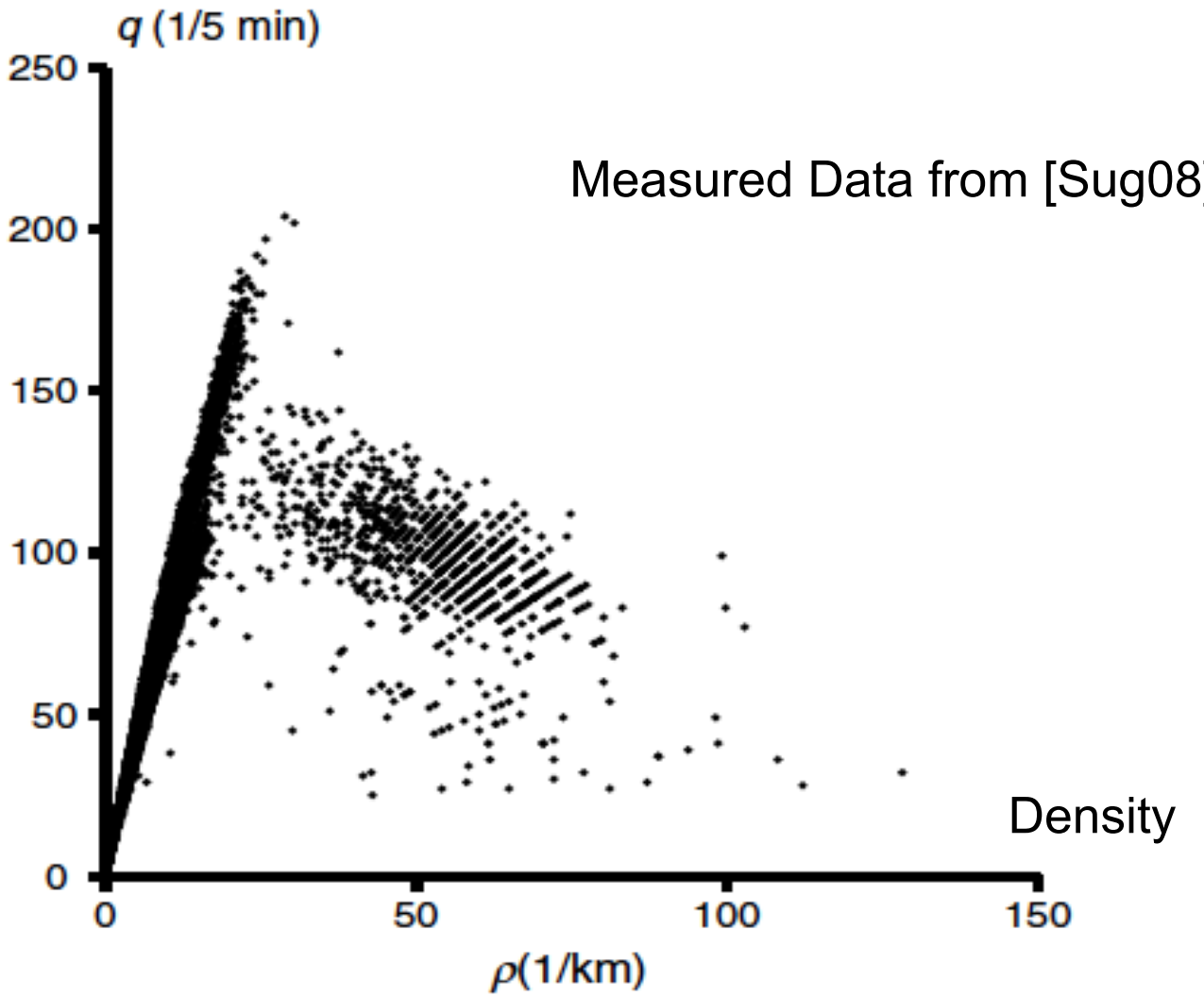
In some cases, the new concept has already been formed in and taken from a domain that is outside the domain that deals with the parts and their interactions.

A substantial revision or extension of the established theory or the development of  a radically new theory is required to explain the occurrence of the emergent property.

# Emergent Composition—Traffic Jam

# Dependence of Throughput on Traffic Density

Throughput




Measured Data from [Sug08]

Density

# Relations between A Level and B Level

- The occurrence of novel qualities at the A level depends on the satisfaction *of enabling conditions* on the B level(*supervenience* of A on B, *upwards causation*).
- Any change of state of A effects the states of B (*downward causation).*
- The phenomena at A are described by models that deal with a *reduced representation* of B.
- In many cases, *new* (relative to B) *concepts* are introduced in the models of A in order to simplify the description.
- The *nomological constraints* of B limit the potential states of the models of A. The processes of A act within these constrained state space.
- In a systems that shows emergence, some of the stochastic processes of B may be replaced by *systemic processes* of A, governed by *novel rules*.

# Supervenience of A on B

- Supervenient properties are macro-properties that can differ only if their micro-property bases differ.
- There can be no difference in a supervenient property without a difference in their microbase.
- Different micro-property bases can realize the same macro-property.

# Example: Traffic Jam

- A traffic jam is a property of the whole—this concept does not exist at the level of an individual car.
- Interaction of cars are constrained by the environment (road). If traffic density is above a certain limit (enabling condition) then a traffic jam (property of the whole) is likely to occur
- In light traffic, the slowdown of one car does not result in the slowdown of the car behind. In dense traffic, a following car must slow down if the preceding car slows down (new rule at A—downward causation)
- The formation of a traffic jam can be explained by a augmented theory at the B-level.

# Explanation of an Emergent Property E at A

1. Functionalize E, i.e., find a causal relation between E and other properties of interest—otherwise E is only *epiphenomenal*
2. Find a new model at the level of B that includes E and establishes a relationship between B and E
3. The new model is an augmented base model—it goes beyond the original model describing B

B Set of basal properties and relations

E Emergent property

# *Explained* vs. *Unexplained* Emergence

An emergent property can be *explained* iff there exist a theory that contains in its domain

- the characterization of the parts
- the characterization of the interactions, including their timing.
- the concept that captures the emergent property

and predicts the conditions when the emergent property will occur.

If such a theory does not exist, the emergent property is currently *unexplained*.

# Summary: Characteristics of *Emergence*

- Distinction between Base (Micro) level and Abstract (Macro) level
- Interactions of components in physical or cyber space
- Birth of novel phenomena (properties, structure, behavior, patterns etc.) -- Wholism
- Temporal dimension (dynamic)
- Enabling condition at the micro-level
- Upward causation
- Downward causation
- Supervenience of A-level on B-level