On the 3\textsuperscript{rd} Dimension of Data

Reginald Cushing

Informatics Institute
University of Amsterdam
Introduction

• Data is increasingly becoming mobile, heterogeneous, morphic, transient.
  – Mobile: in continuous motion e.g. over networks.
  – Heterogeneous: having different varieties.
  – Morphic: changing, transforming into new data.
  – Transient: existing for short span of time.

• Yet many of our systems still treat data as static.
  – Data in a database/fileserver/archive

• How do we capture such dynamicity of data?
Dimensions of Data

Static structure: Types, OO, OWL

Semantic structure: RDF, production rules

Temporal structure: Automata, Petri-nets
Dimensions of Data

Static structure: Types, OO, OWL

Semantic structure: RDF, production rules

Temporal structure: Automata, Petri-nets

- car
- 4 wheels
- steering
Dimensions of Data

Static structure: Types, OO, OWL

Semantic structure: RDF, production rules

Temporal structure: Automata, Petri-nets

- car
  - 4 wheels
  - steering

- car used by human
  - human exist on Earth

- car exists on Earth
Dimensions of Data

Static structure: Types, OO, OWL

Semiotic structure: RDF, production rules

Temporal structure: Automata, Petri-nets

**Dimensions of Data**

- **Static structure**: Types, OO, OWL
  - car
    - 4 wheels
    - steering

- **Semiotic structure**: RDF, production rules
  - forge
  - metal
  - scrap
  - garden furniture

- **Temporal structure**: Automata, Petri-nets
  - car used by human
  - human exist on Earth
  - car exist on Earth

- Car used by human
- Human exist on Earth
- Car exist on Earth

- Car on Earth
- Garden furniture
- Scrap
- Metal
- Forge
Automata - temporal structure

- Model asynchronous concurrent event systems.
  - Shared action model vs shared memory vs message passing.
- Allows us to describe data as it changes/gain knowledge over time.
- Allows us to reason about the temporal complexity of data.
  - Number vs LHC image.
- Allows the decomposition of data processing into transition states.
- Act as the bases of a concurrency coordination model for data.
Uses – distributed computing-1

• Describe data processing as an automaton on a data object.
  – Data object: image, text.
  – Automaton: states object can be in.

• Distribute transition functions & coordinate data state transitions.

• Listen for relevant data states to capture output.
Uses – distributed computing-2
Uses – distributed computing-3

Data objects d are transitioned from one state to the next using an automaton to guide the transitions which are done using d-op. Data is processed along the way thus $d \rightarrow d' \rightarrow d'' \rightarrow d^k$ where $d^k$ is the $k^{th}$ derivate of the data object.
Uses – distributed computing-4
Uses – data storage

• Making data in stores active.
  – Databases, fileservers, data warehouses, etc

• Giving data temporal structure such as different possible states of stored files one can query different states of the stored data.
  – Invoking back-end processing where no data for a state exists.

• Storage systems can optimize what to store.
  – We only need to store data once if we have a bijective mapping between 2 sets of data.
Uses – many cores

• The tendency in computer processor architecture shows that future processors are to have many cores.

• Such architectures present a challenge to programmers as to how best exploit the full potential of so many cores.

• Abstract data models can aid in decomposing and coordinating the data processing in memory on many cores.
Microprocessor Transistor Counts 1971-2011 & Moore's Law

The curve shows transistor count doubling every two years.
Book analogy - 1

I’m Marvin, I can process books.

Find the scientific English books and convert them to Audio books.
Book analogy - 2

Now we want to convert English books to Dutch....Marvin does not know how to do this.
- We can update Marvin to know everything again.
- We can have specialized Marvins that can do one thing well e.g. recognize English books:
  - I know English
  - I recognize Science books
  - I can convert English to Audio
  - I can convert Dutch to Audio
  - I can translate Eng-Dutch
Book analogy - 5

Book temporal structure used to coordinate communication between cores. (shared events)
Ownership transferable memory based on temporal structure. 0-copy.
Data-centric Computing on Distributed Resources

Reginald Cushing

http://dare.uva.nl/record/1/486948