Novel Software Engineering Method & System for Neuromorphic Computing & Machine Learning

백 옥기
白 鈺淇
O.K. Baek

The IT industry has come a long way and made a significant, for hardware

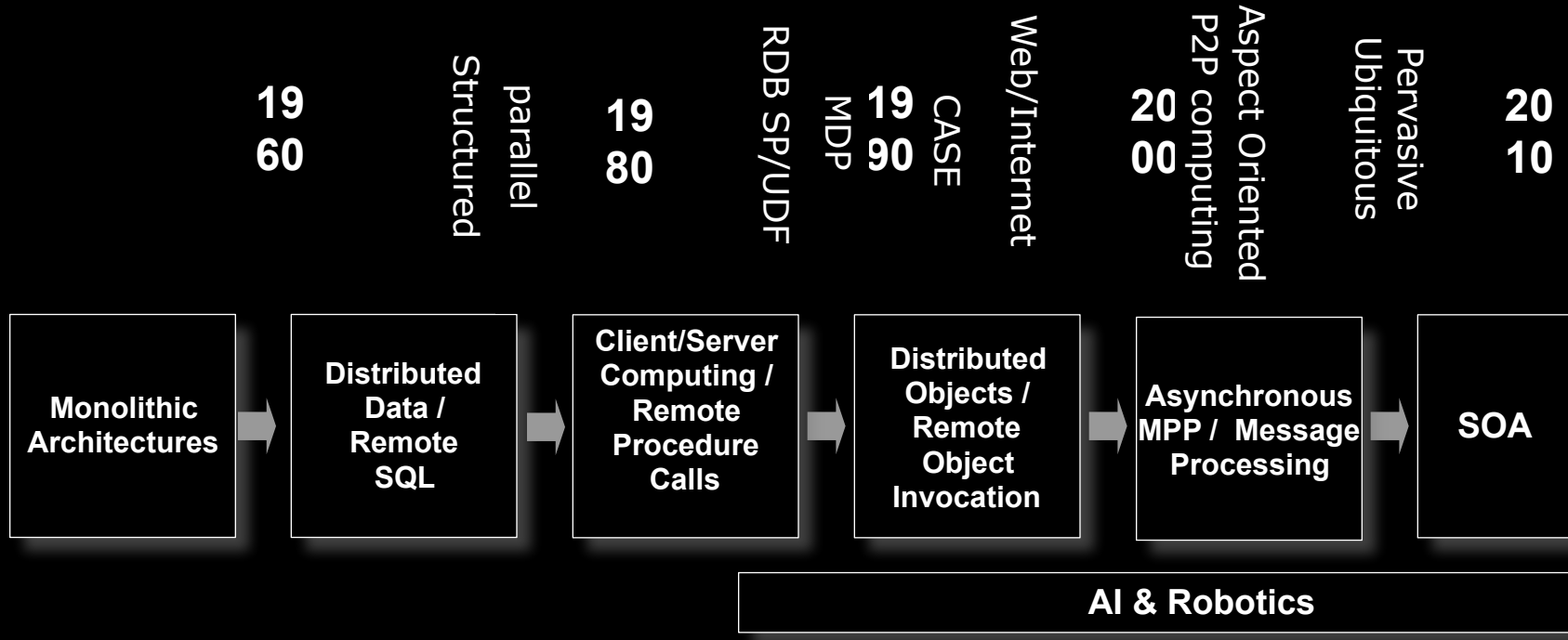
PetaFLOPS

GigaFLOPS

MIPS

KIPS

The Computing Model and Software Engineering Method stayed more or less same for the 5 decades and very little progress has been made so far



Autonomous Computing
Cognitive Computing
Deep Learning

Object-Oriented Programming and Service-Oriented Architecture are not new.
OO is to enable reuse; SOA is for Modularity and Loose Coupling.

- Reuse
  - FORTRAN, Ada, PASCAL subroutines
  - COBOL Copybooks
- Modularity and Loose Coupling
  - Parallel FORTRAN
  - C++ Virtual Functions
  - Ada Rendezvous

We've made admirable progress in terms of computing power & hardware size.

The architecture stayed the same: Von Neumann Architecture i.e., Calculator

IT industry has been focusing on processing human-initiated events and human-generated data

- Human generated data
  - linear
  - numeric data
  - structured data
  - low velocity
  - low volume
  - at-rest

IT industry has been focusing on processing human-initiated events and human-generated data

- Human initiated events
  - linear
  - deterministic
  - anticipated
  - low velocity
  - less time critical

IT industry has been focusing on processing human-initiated events and human-generated data

Finite combinations and permutations

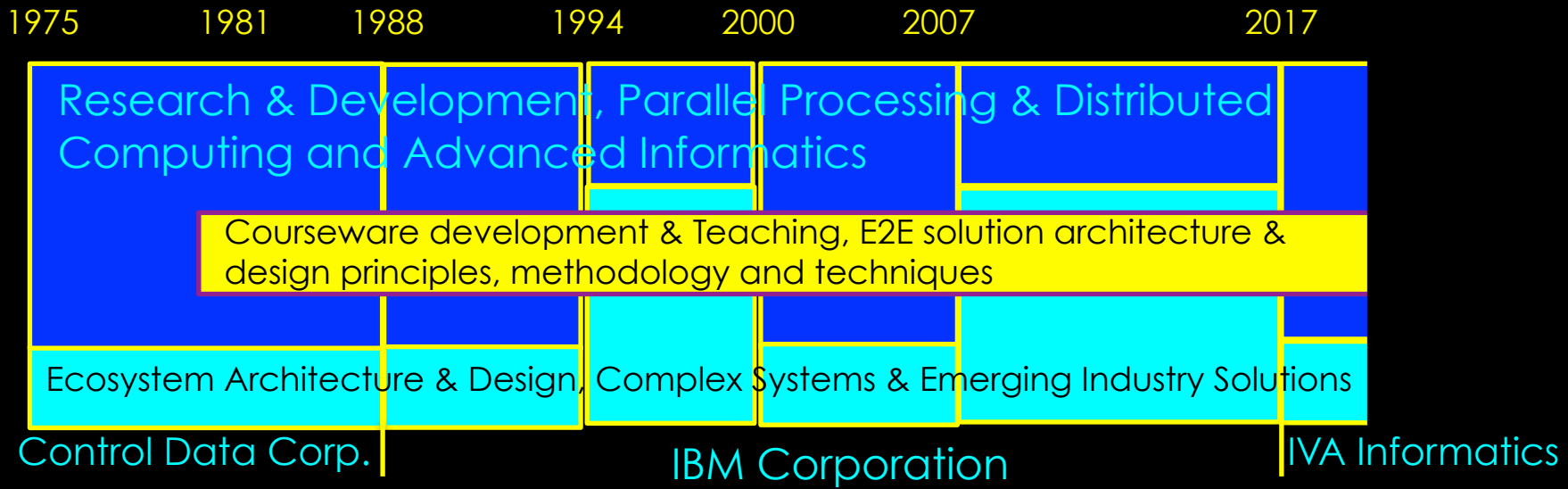
We can get prepared for anticipated events and exceptions.

We need an Intelligent machine that thinks and learns itself to handle unanticipated events and self-adapts to various changes!


We have a very, very long way to go.

Who am I?

OK Baek, 백 옥기



Executive Architect, IBM Corporation
Senior Technical Staff Member, IBM Corporation
Chief Solutions Architect, IBM Global Services
Senior Research Staff Member, IBM Research
Master Inventor, IBM Corporation
Certified Instructor, IBM Corporation
Global Solutions Executive, IBM Life Sciences Group
Canadian National Delegate to ISO/IEC JTC1/SC21 WG6
Chief Systems Architect, IBM Software Group
Senior Systems Engineer, Control Data Corporation
Certified Lecturer, Control Data Institute

Information Science, McGill U
Bioinformatics, MIT
Neuroscience, McGill U
Molecular Biology, McGill U
Computer Science, U of California, U of Toronto
Computer Engineering, U of Minnesota, Control Data Institute
Electronics, SNU

Who am I?

OK Baek, 백 옥기

Executive Architect, IBM Corporation
Senior Technical Staff Member, IBM Corporation
Chief Solutions Architect, IBM Global Services
Senior Research Staff Member, IBM Research
Master Inventor, IBM Corporation
Certified Instructor, IBM Corporation
Global Solutions Executive, IBM Life Sciences Group
Canadian National Delegate to ISO/IEC JTC1/SC21 WG6
Chief Systems Architect, IBM Software Group
Senior Systems Engineer, Control Data Corporation
Certified Lecturer, Control Data Institute

Who am I?

OK Baek, 백 옥기

Information Science, McGill U
Bioinformatics, MIT
Neuroscience, McGill U
Molecular Biology, McGill U
Computer Science, U of California, U of Toronto
Computer Engineering, U of Minnesota, Control Data Institute
Electronics, SNU

IT industry needs to process machine-generated real-time data and environmental events as well

Machine generated or Environmental data

- nonlinear
- image data
- unstructured data
- high velocity
- high volume
- in motion

Environmental events

- nonlinear
- indeterministic
- unanticipated
- high velocity
- time critical

Infinite combinations and permutations

We do not know what we do not know, therefore cannot predict and get prepared.

IT industry needs to process machine-generated real-time data and environmental events as well

Machine generated or Environmental data

- nonlinear
- image data
- unstructured data
- high velocity
- high volume
- in motion

IT industry needs to process machine-generated real-time data and environmental events as well

Environmental events

- nonlinear
- indeterministic
- unanticipated
- high velocity
- time critical

IT industry needs to process machine-generated real-time data and environmental events as well

Infinite combinations and permutations

We do not know what we do not know, therefore cannot predict and get prepared.

We need a New Computing Model and a New Software Engineering Method:

Data-centric computing model
as opposed to Method-centric model;

Image processing
as opposed to "computing" or text processing;

Dynamic software component
as opposed to Static software component;

Adaptive software
as opposed to preset software;

Self-learning software based on "experience";
etc.

Neurosynaptic & Neuromorphic computing Chip

Brain-inspired Computer
Supercomputer on a chip

non-von Neumann

Motivation

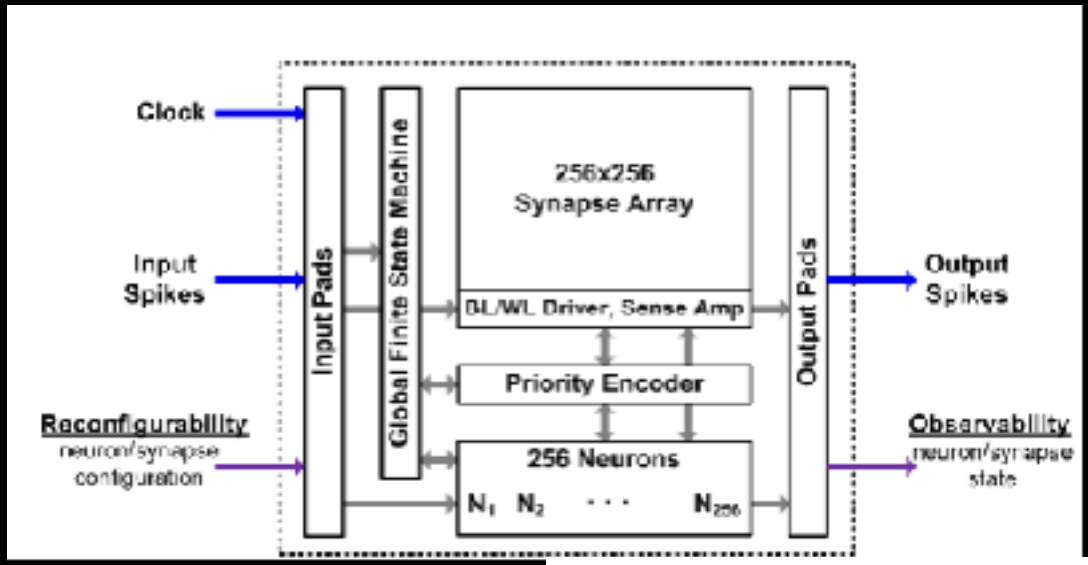
DARPA SyNAPSE (Systems of Neuromorphic Adaptive Plastic Scalable Electronics) Project
Supercomputer of size of postage stamp
45nM SOI-SMOS chip
10^{10} neurons and 10^{14} synapses
in <2L space running
at < 1KW power consumption
46 billion synaptic operations per second
Dynamic reconfiguration
Intuitive natural programming paradigm

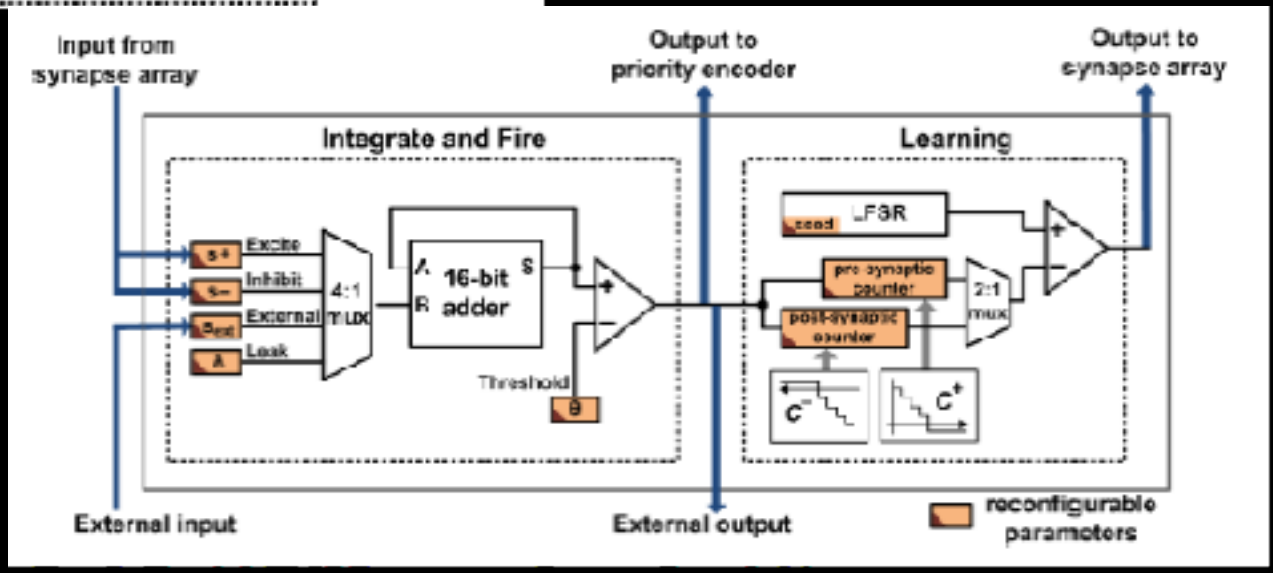
Motivation

- Real Time processing
- Fault tolerant
- Non von-Neumann parallel
- Low power
  - 20 mW/cm^2
  - 70 mW in operation

- Event driven
- Parallel
- Distributed
- Scalable
- Intuitive human-machine interface

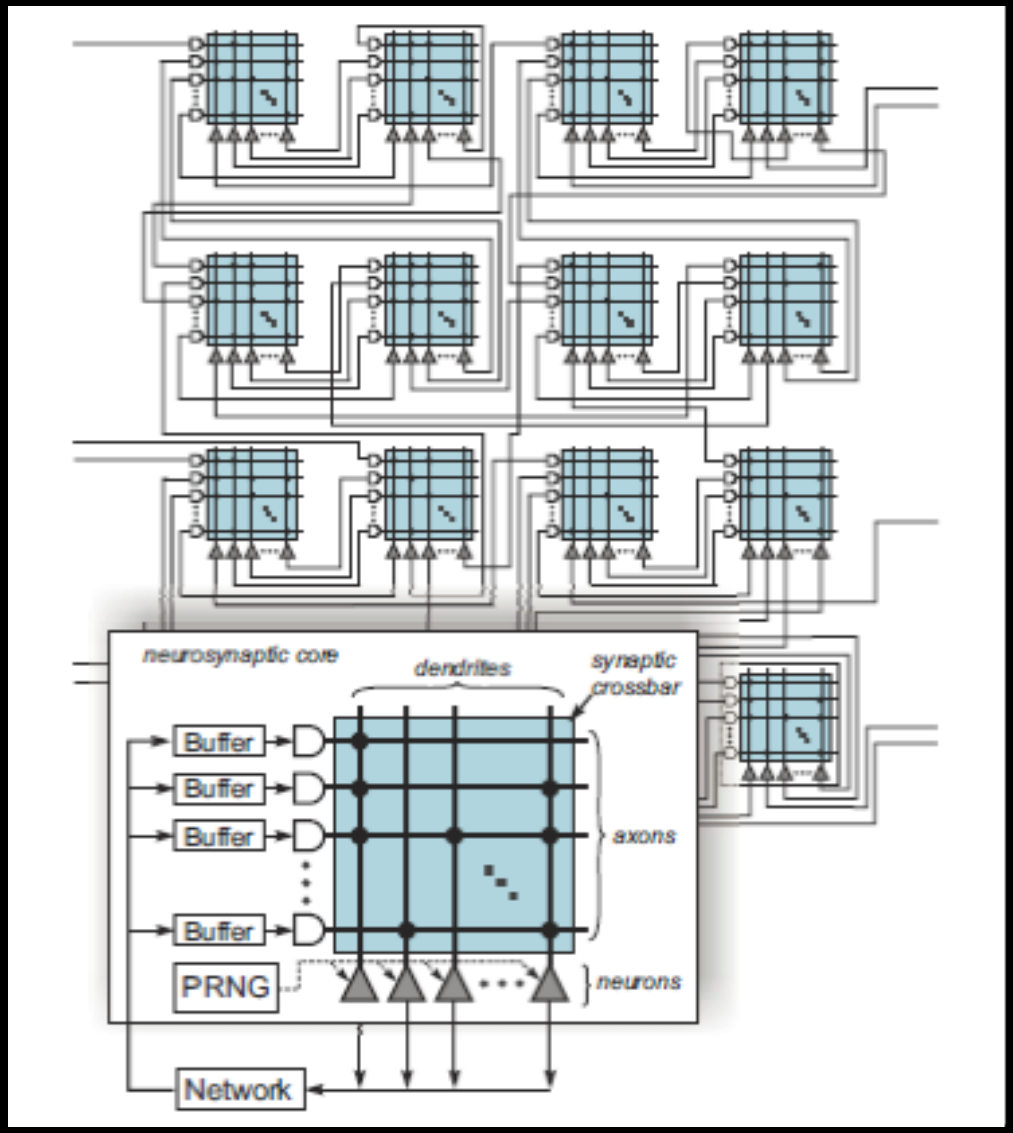
Machine Learning on a chip





Plasticity via reconfigurability

Neurosynaptic Core



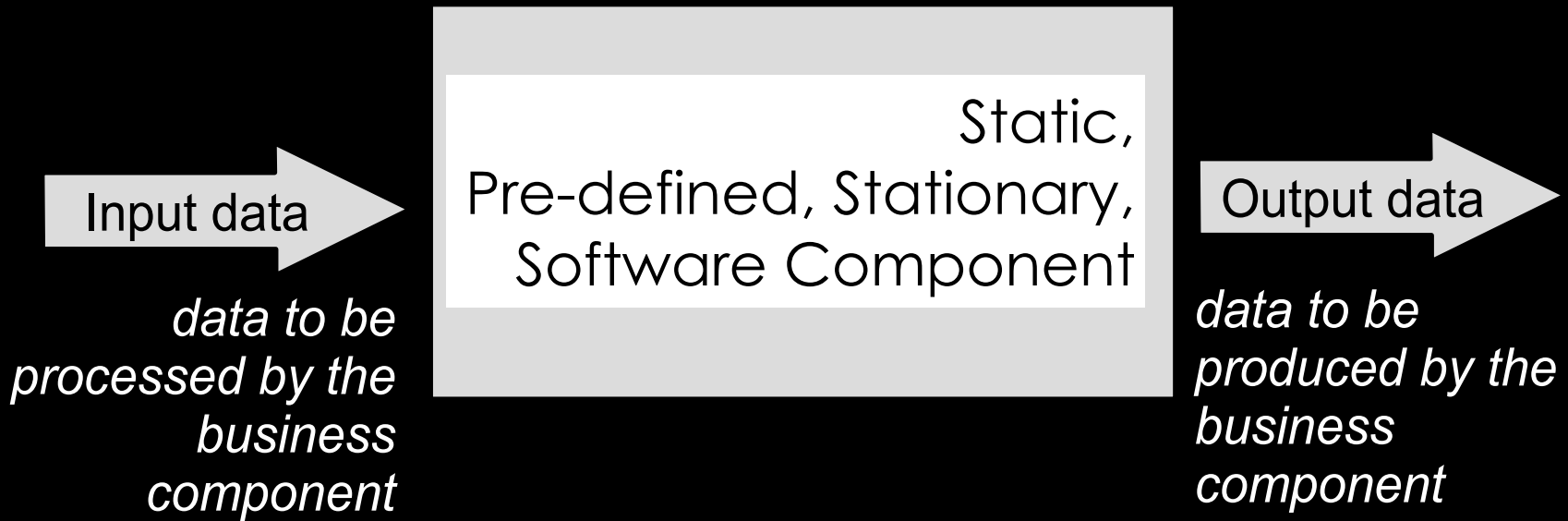
True North Chip
45nm SOI-CMOS
neuromorphic chip

Spike-based message-passing network of:

- 256 neurons
- 256 axons
- 256 x 256 synapse crossbar

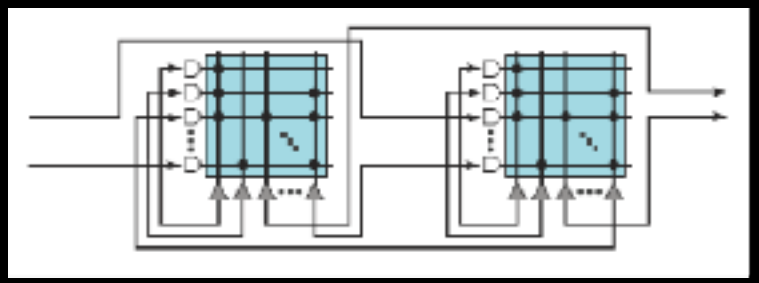
PRNG: Pseudorandom Number Generator

Traditional Software Engineering Model is inadequate and inefficient for IT in the Information Age



Output data = Function (Input data)

Software Component needs to be self-adaptive to unanticipated events



Input data

data to be processed by the business component

Business Component: externally configurable generic set of business functions

Output data

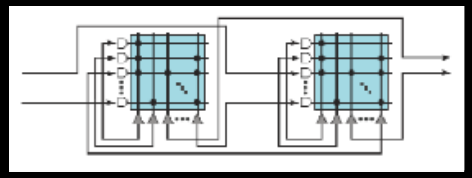
data to be produced by the business component

Control

Business rules to be used by the business component

Output data = Function (Input data, Control input)

Novel Software Engineering Method & System for Dynamic, Self-Adaptive, Itinerant Software migrating to Data and adapting to changes in business and operating environment



Input data

input data and observed events

Dynamic, Self-Adaptive, Itinerant, Software Component

Output data

associations, correlations, co-variances trends

machine learning base on earlier output & context, i.e., experience

Control

Context, domain and ontology business/environmental parameters

Output data = Function (Input data, Control input)

Novel Software Engineering Method for Machine Learning via Plasticity

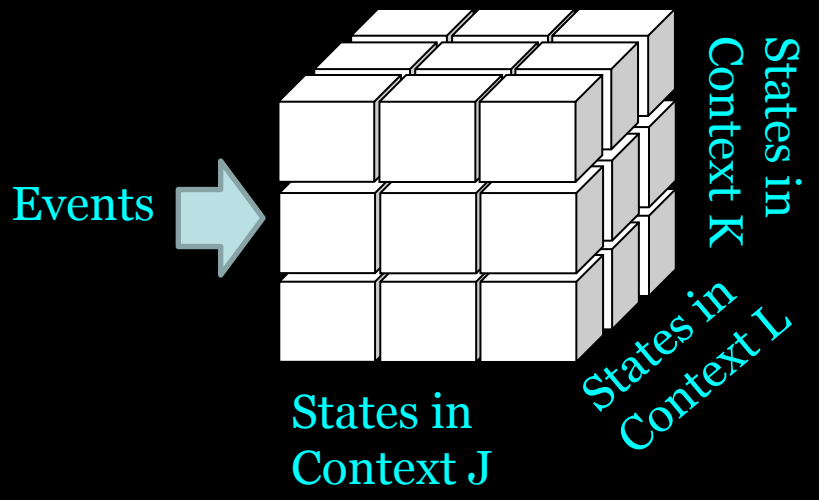
Traditional Model

```
IF (Y or Z) & (A&B&C or D&E)
    IF ...
        IF ...
            IF ...
                IF ...
                    IF
                        do 1, do 2,  ...
                    ELSEIF ....
                            case

ELSEIF K & ...
```

Context-aware Multidimensional Dynamic FSM

F(x) = mdFSM (e1,e2,... s1,s2,... t+n)
Perform F(x);



dynamically update the FMS for newly discovered events and also based on the outcomes of earlier actions

There's always a way to do it better.
Let us find it.

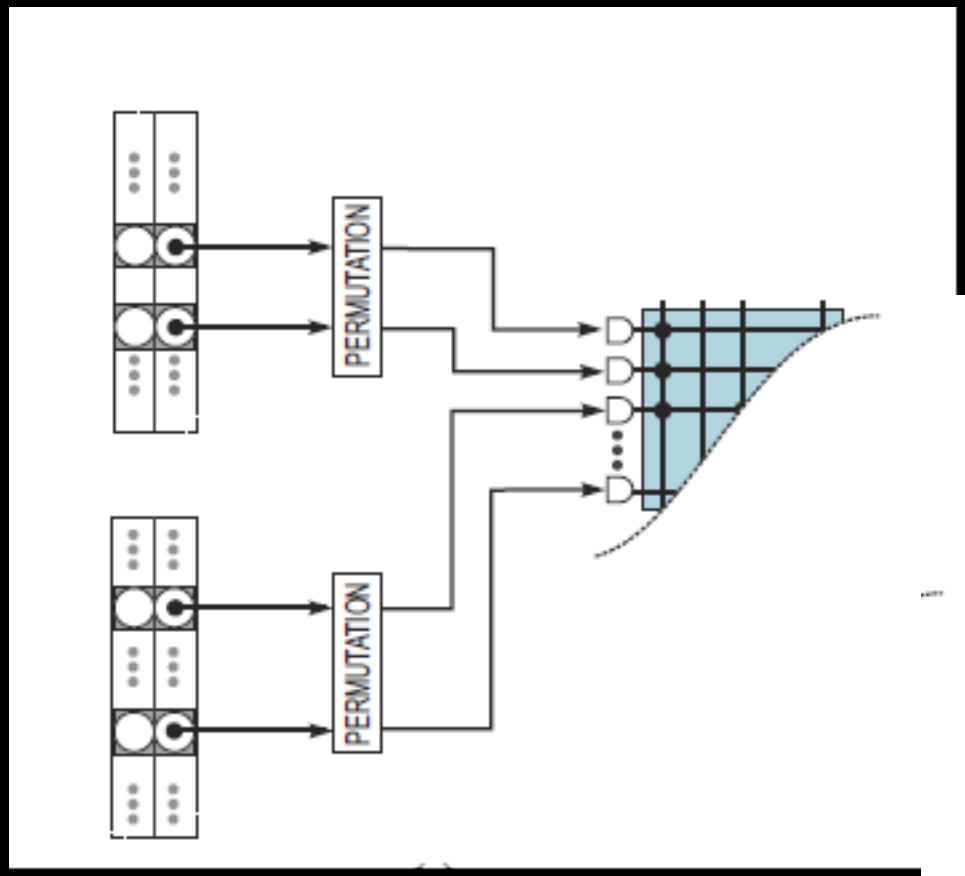
– Thomas Edison

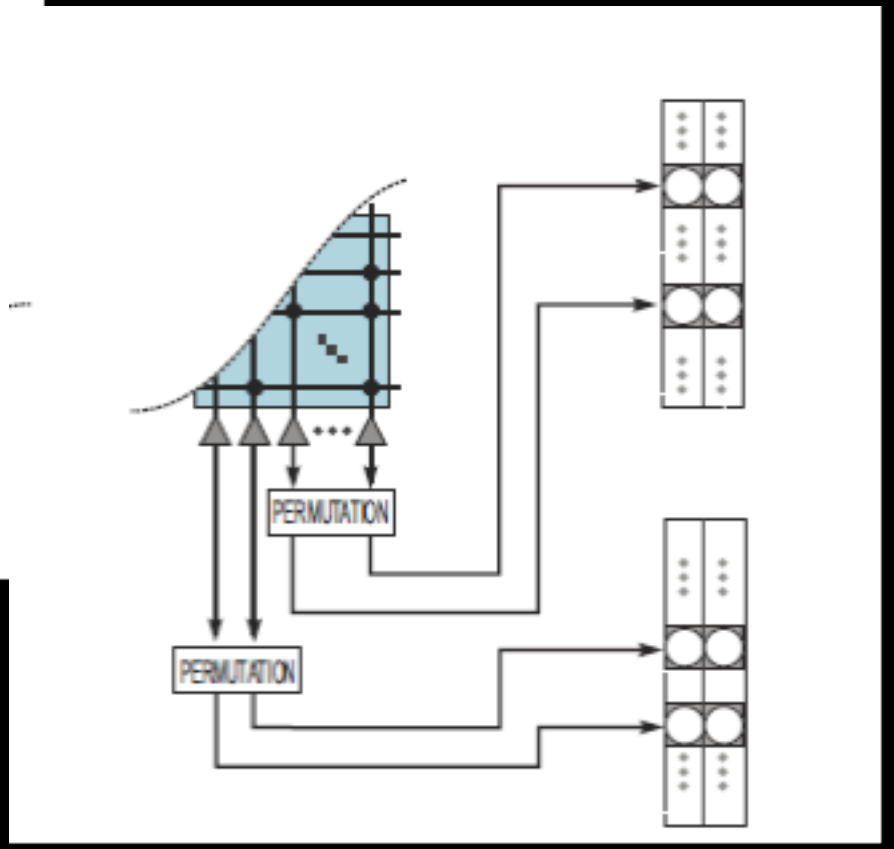
It's easier to come up with
new ideas than to let go of old ones.

- Peter Drucker

대단히 감사합니다

Multiple input for spikes and configuration





Corelet and Programming Language