Electronics

An introduction for Architects

Architectural Association London
Wednesday 2nd May 2007

Chris Leung RIBA

Sponsored by: Haque Design Research Limited
http://www.haque.co.uk

Background

- I studied architecture at UCL
- I am currently doing doctoral research sponsored by Haque Design Research Limited
- Developing an open web-based resource for exchanging environmental data between anyone, anywhere at anytime called: ‘Environment-XML’
- Developing applications for an imaging-based technology to capture the performance of passive and low-energy actuator systems for architectural applications

Content

- Introduction with some basic theory
- Introduction to a systems approach
- System design issues
- Software design issues
- Discuss practical implementation issues
- Q&A
- Demonstrations and Browse

Some theory

- What is Electronics?

  ‘manipulates electrical energy as it passes between different mediums and materials, usually from a state of high potential difference to a state of lower potential difference’

Basic quantities

- Potential difference (or p.d.) (V), units: Volts
- Current (I), units: Amperes
- Resistance (R), units: Ohms
- Their relationship given by:

  $I = V / R$ (Ohms Law)

Basic concepts

- Electromotive Force (EMF) drives electrical current from a high potential difference (p.d.) to a low p.d., the amount of current that flows is related to the amount of electrical resistance of the material between two points of p.d.
- Electronics is about marshalling p.d. to do the things you want to do, when you want them done, in the manner you would like.

Systems Approach

A systems approach can be applied to electronics to decompose it into;

- Modules for inputs (e.g. switches, sensors)
- Modules for processing (e.g. a controller)
- Modules for outputs (e.g. a lighting dimmer)

Inputs

- An input module in your system triggers or changes the behaviour of the flow of electrical energy.
- An electronic sensor is a type of input that includes a transducer that converts energy from a physical medium into electrical energy or changes the behaviour of electrical energy in a circuit, e.g. an Light-Dependent-Resistor (LDR)

Processing

- A process module takes the state of the input and introduces a measure of control on the output
- e.g. A thermostat is a type of process module that controls an output often between a high and low set-point, the state of the output is controlled by measuring the state of the input with respect to the set-points

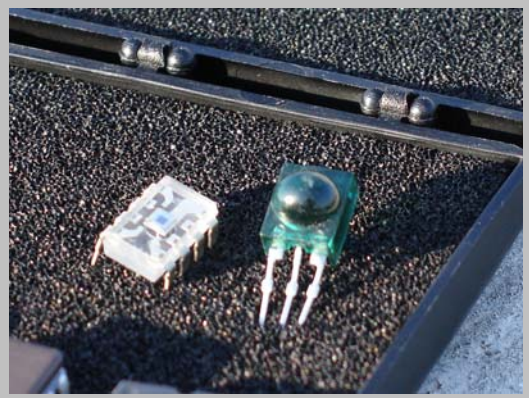
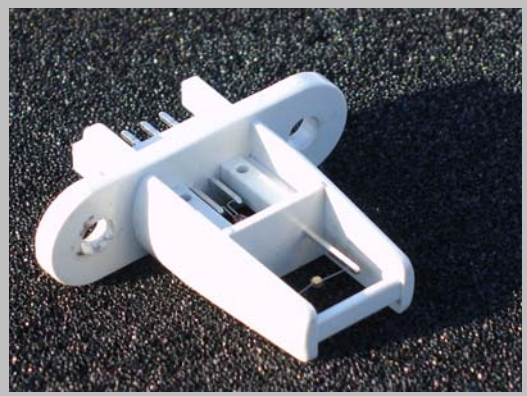
Outputs

- The output module affects the external world in some way by translating electrical energy into some useful work
- e.g. a loudspeaker converts electrical energy into sound waves propagated through air

Input modalities

Input modalities

- Environment - Physical
  - Temperature e.g. digital thermometers, iButton thermochron
  - Relative Humidity e.g.
  - Wind e.g. weather station suite
  - Gas composition/concentration of Air

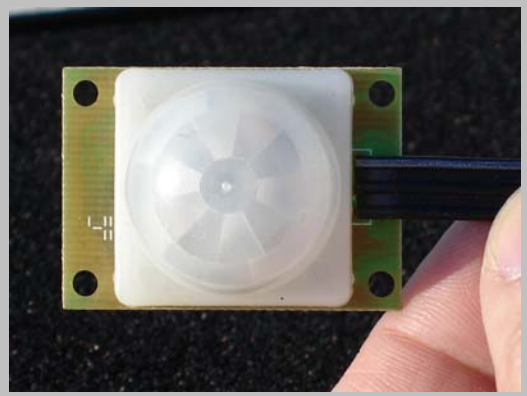




Input modalities

- Environment - Electromagnetic radiation
  - Radiowaves e.g. antennae
  - Microwave e.g. inductors
  - Solar/UV radiation e.g. pyranometers
  - Visible e.g. LDR’s, colour sensors

Input modalities

- Movement
  - Linear, rotation, e.g. potentiometers, optical encoders
  - Rotation rate, e.g. digital tachometer
  - P.I.R.s passive infra-red sensors
  - Imaging e.g. USB WebCam, CCD Video camera

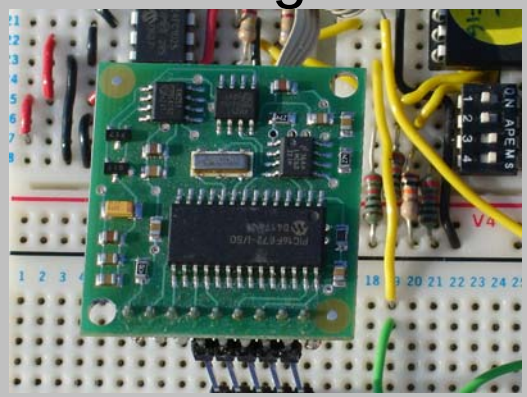


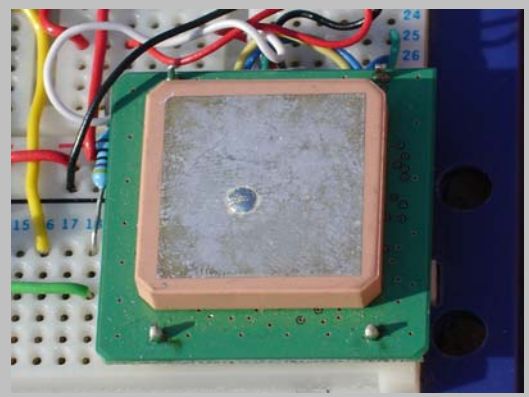
Input modalities

Movement

- Presence e.g. switches, optical beam-break sensors, pressure-pad mats, capacitive touch sensors
- Through Fields e.g. magnetic ‘Hall-effect’ sensors

- Position
  - e.g. position encoder, GPS receiver, inclination sensor, compass heading





Input modalities

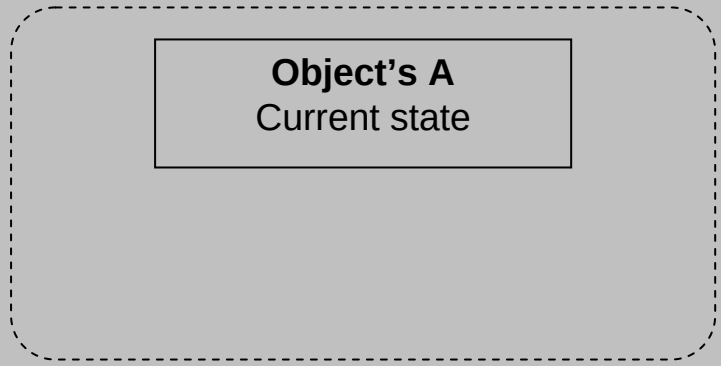
- Human
  - Stress e.g. GSV galvanic skin response
  - Heart-rate e.g. medical monitoring
  - Pedometers e.g. walking rate

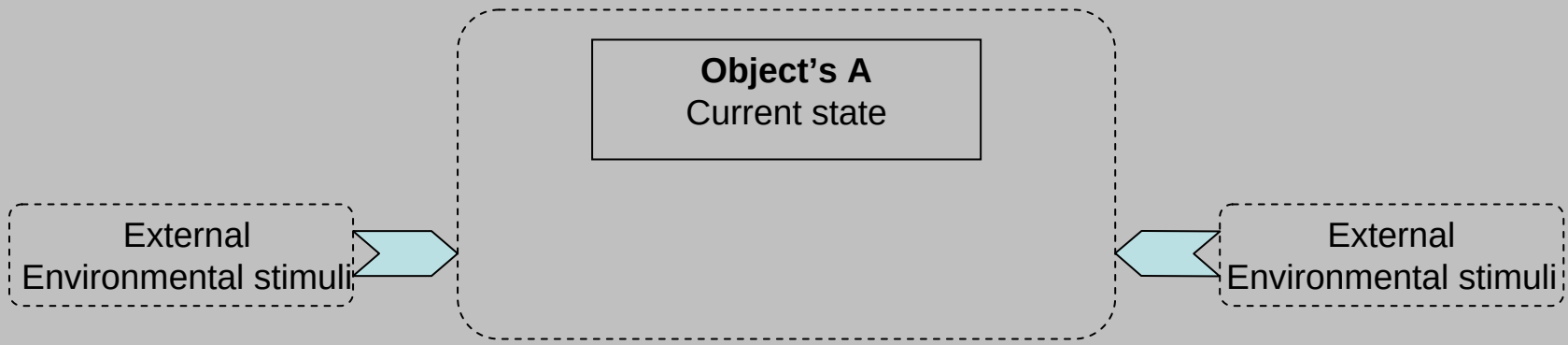
Processing Strategies

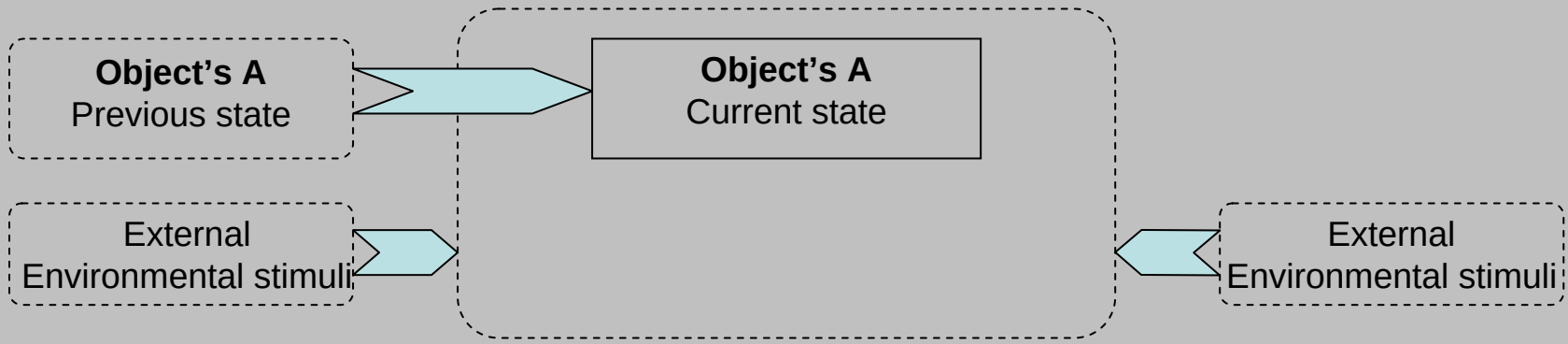
- Logic in hardware
  - Analog and digital logic
- Logic in software
  - Deterministic e.g. state-machines
  - Fuzzy logic e.g. probabilistic

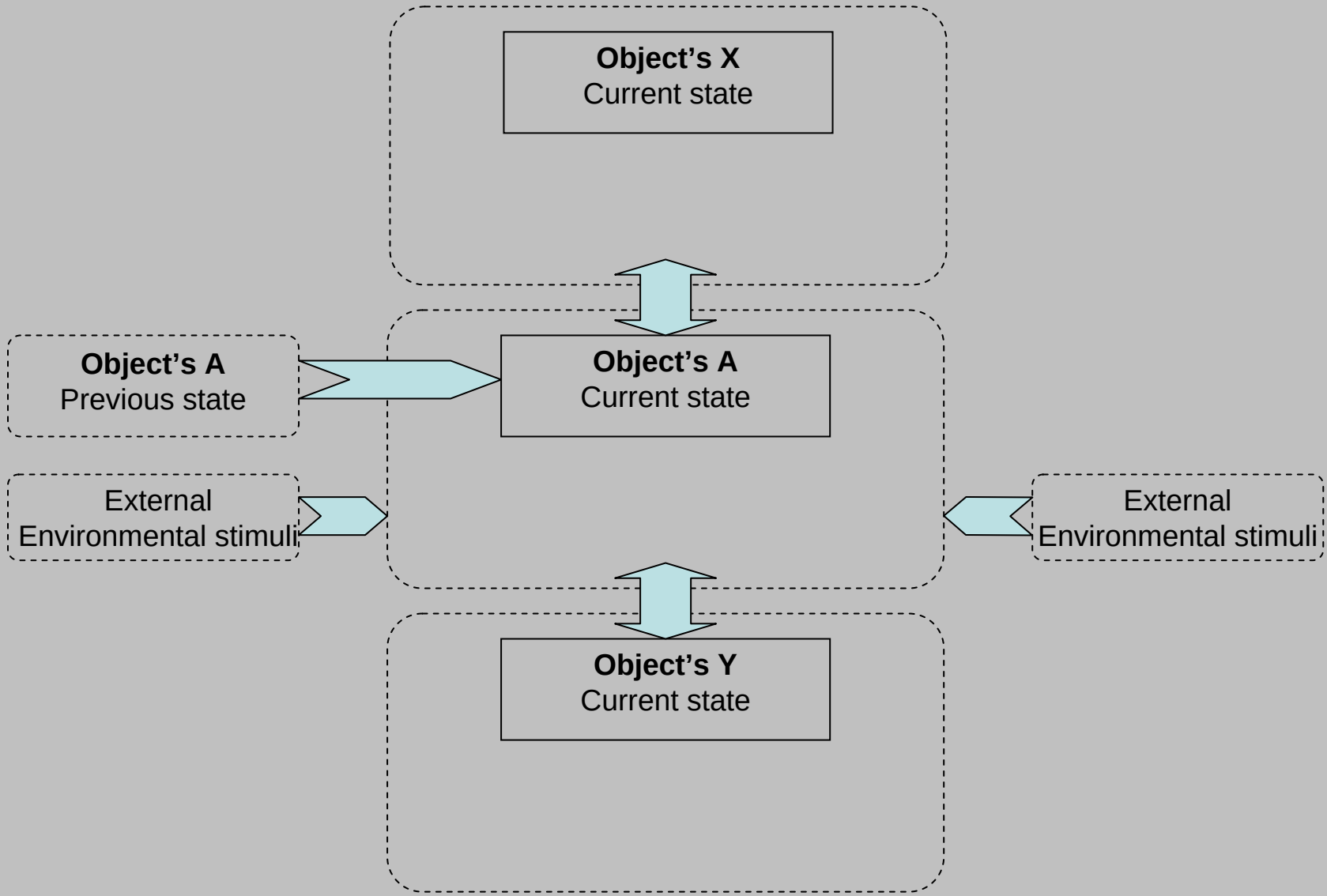
Software Design

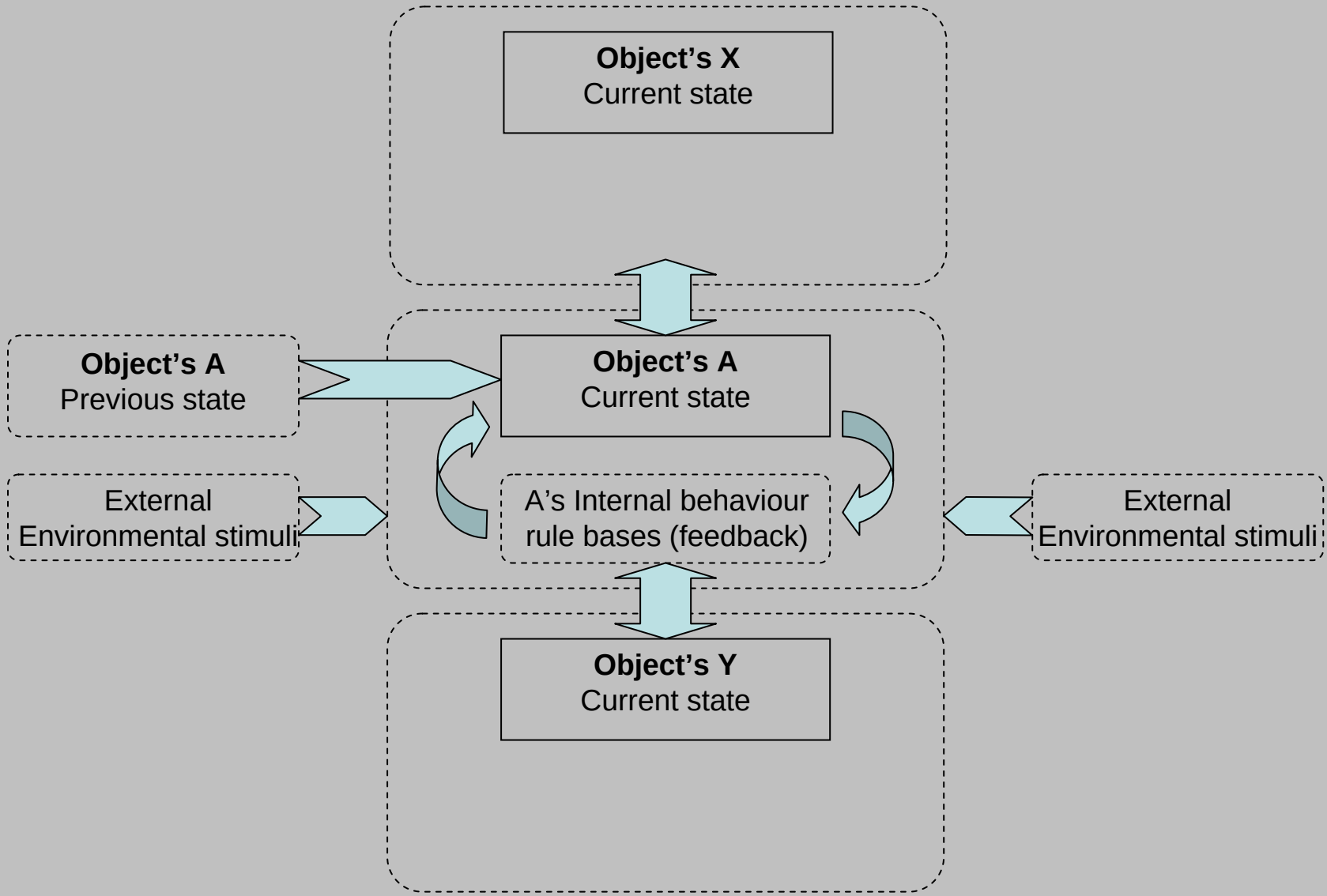
- Do you need to?
- State-Machine approach

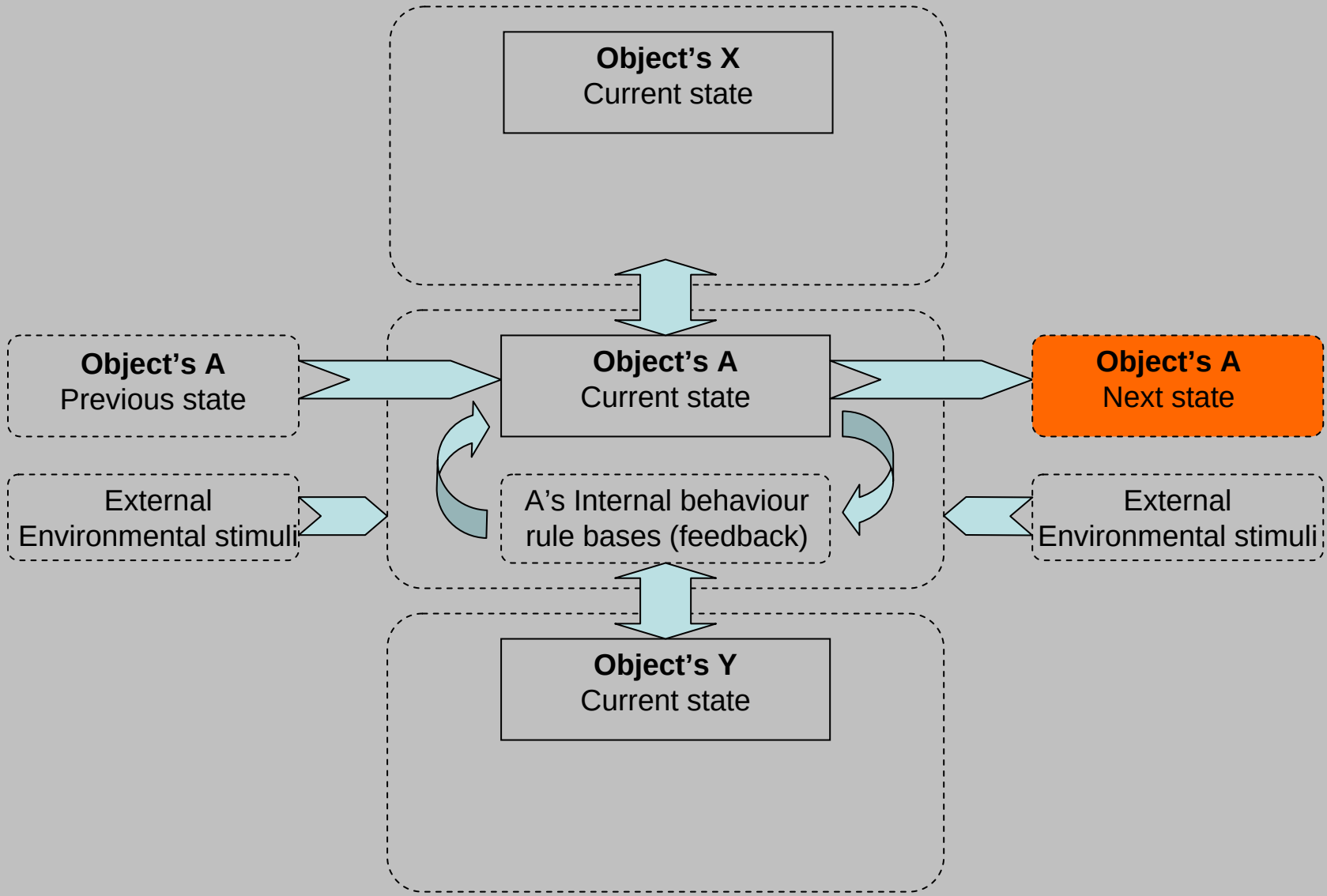

Object's A
Current state

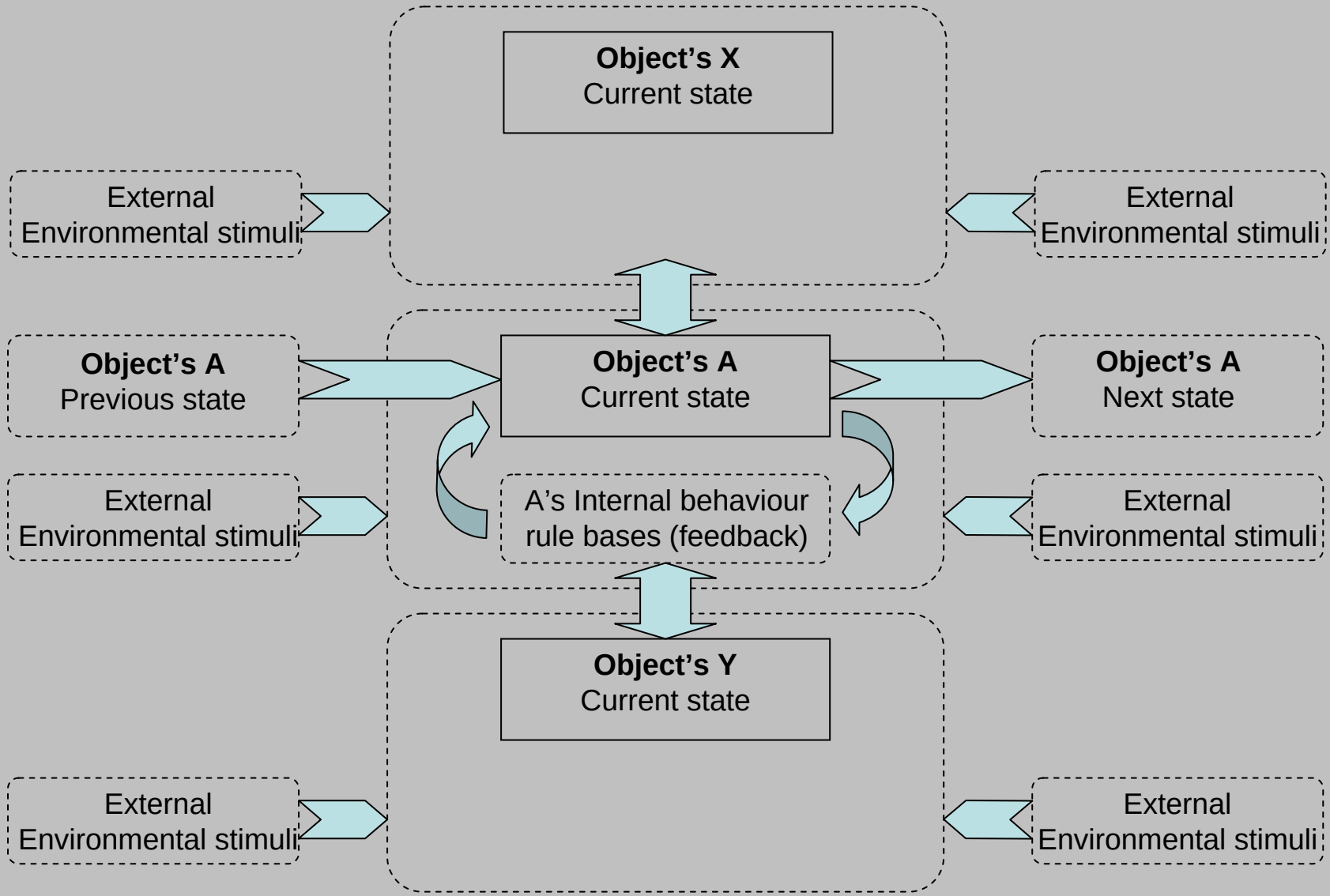

Object's A
Current state
External
Environmental stimuli
External
Environmental stimuli

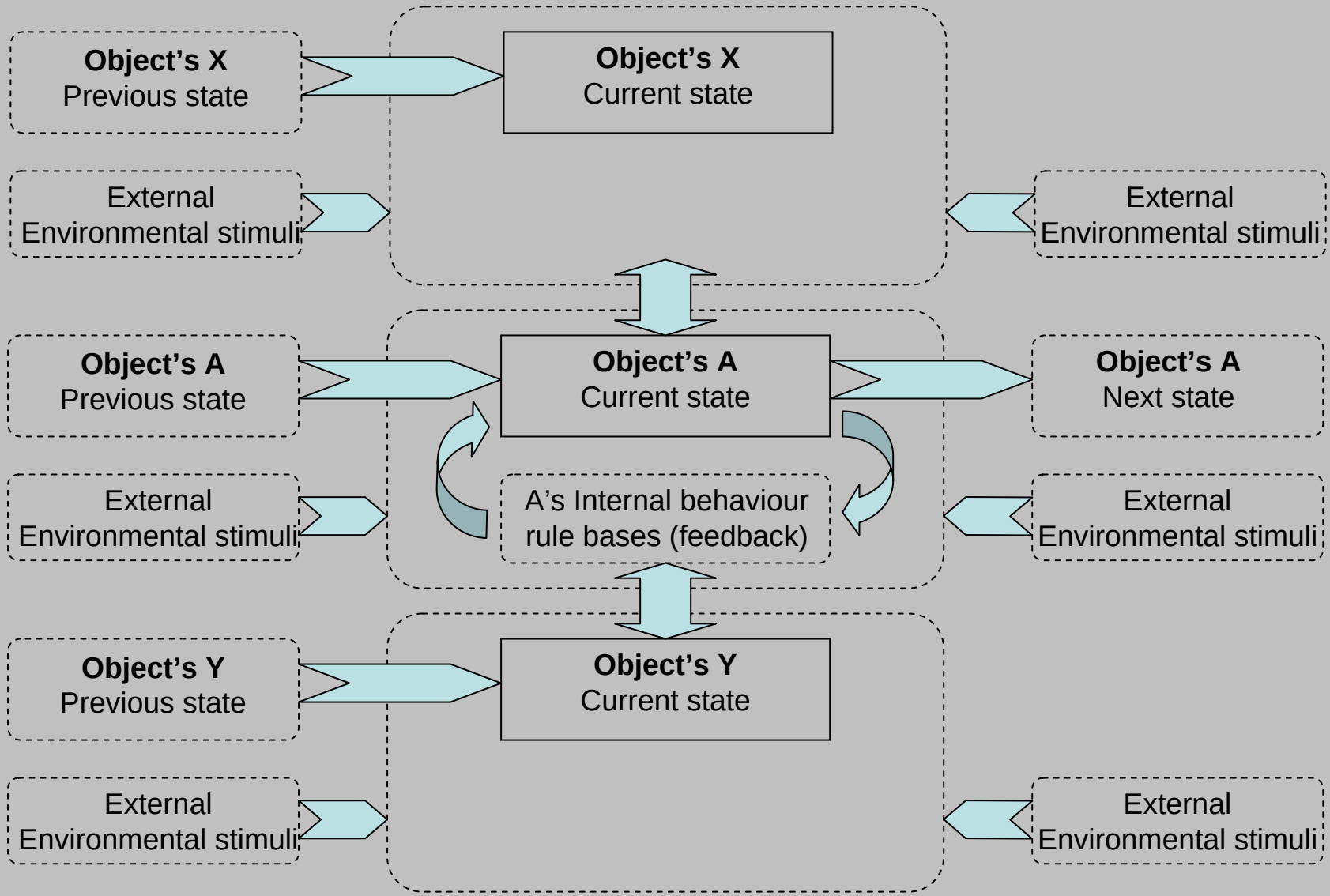

Object's A
Previous state
Object's A
Current state
External
Environmental stimuli
External
Environmental stimuli

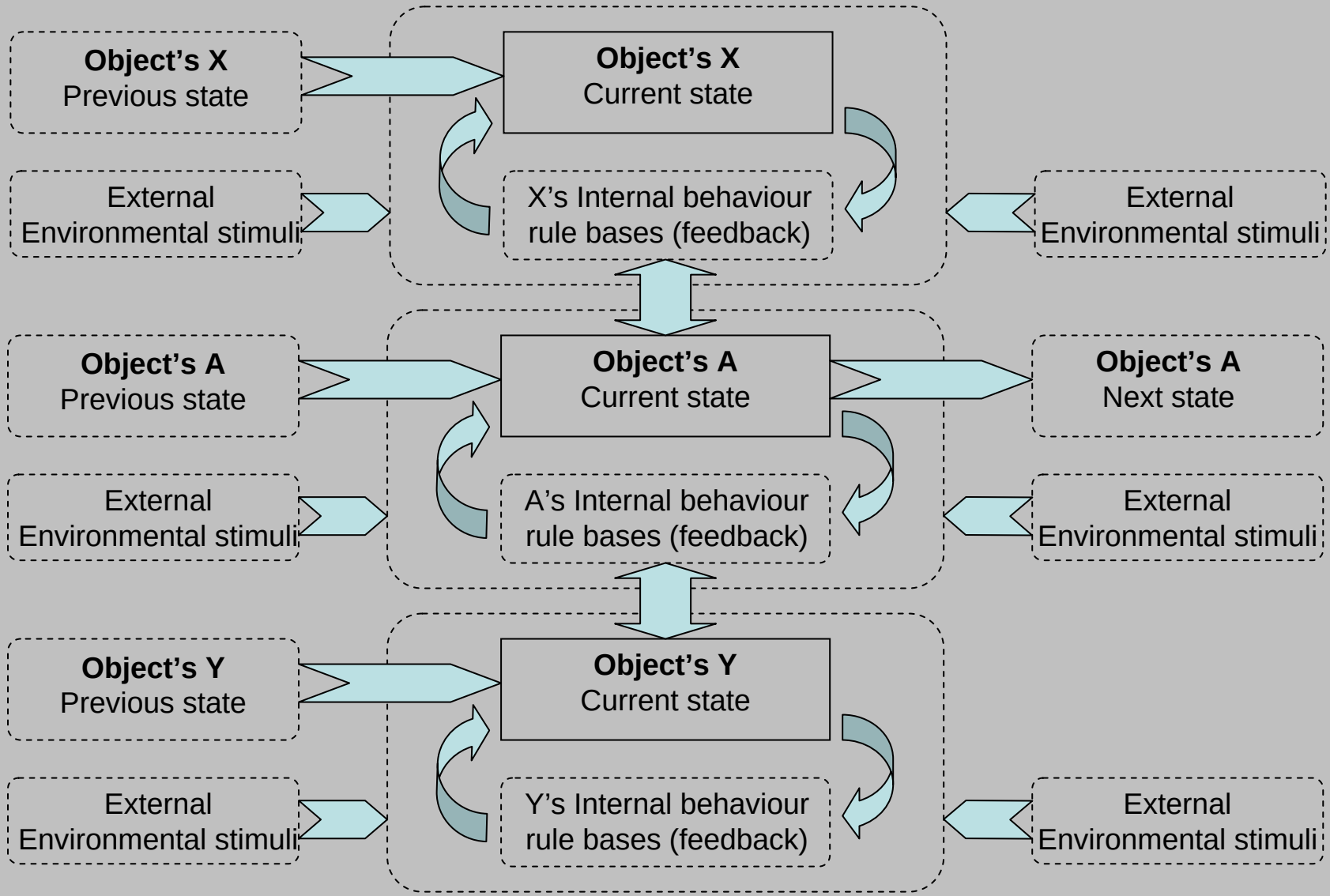

Object's X
Current state
Object's A
Previous state
Object's A
Current state
External
Environmental stimuli
External
Environmental stimuli
Object's Y
Current state

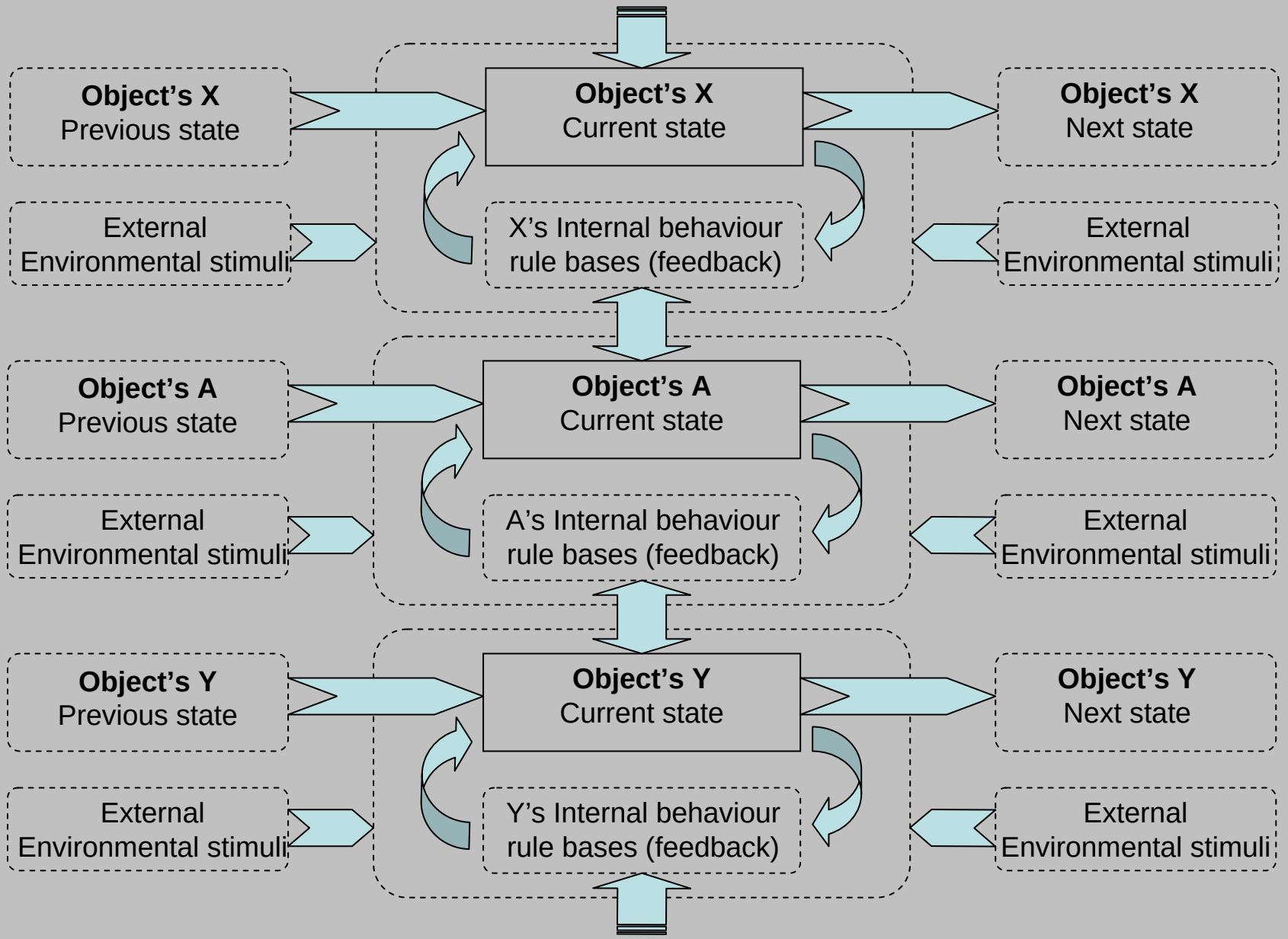

Object's X
Current state
Object's A
Previous state
Object's A
Current state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Current state


Object's X
Current state
Object's A
Previous state
Object's A
Current state
Object's A
Next state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Current state


Object's X
Current state
External
Environmental stimuli
External
Environmental stimuli
Object's A
Previous state
Object's A
Current state
Object's A
Next state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Current state
External
Environmental stimuli
External
Environmental stimuli


Object's X
Previous state
Object's X
Current state
External
Environmental stimuli
External
Environmental stimuli
Object's A
Previous state
Object's A
Current state
Object's A
Next state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Previous state
Object's Y
Current state
External
Environmental stimuli
External
Environmental stimuli


Object's X
Previous state
Object's X
Current state
External
Environmental stimuli
X's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's A
Previous state
Object's A
Current state
Object's A
Next state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Previous state
Object's Y
Current state
External
Environmental stimuli
Y's Internal behaviour
rule bases (feedback)
External
Environmental stimuli


Object's X
Previous state
Object's X
Current state
Object's X
Next state
External
Environmental stimuli
X's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's A
Previous state
Object's A
Current state
Object's A
Next state
External
Environmental stimuli
A's Internal behaviour
rule bases (feedback)
External
Environmental stimuli
Object's Y
Previous state
Object's Y
Current state
Object's Y
Next state
External
Environmental stimuli
Y's Internal behaviour
rule bases (feedback)
External
Environmental stimuli

Output Design

- Light, movement, sound, tactile, spatial modifiers, responses in the virtual

Interaction System design

- http://www.makingthings.com
- http://www.parallax.com

Programming

- Choice of Languages
- Editors, Compilers and Assemblers

Programming

- High-level Languages
  - MaxMSP e.g. Teleo
  - Java e.g. Arduino, Javelin, TINI
  - Basic e.g. Stamp, PIC Micro
  - Spin e.g. Parallax Propeller

Programming

- Editors, Compilers and Assemblers or all combined into an “integrated development environment” or IDE
  - MaxMSP e.g. ?
  - Java e.g. Wiring IDE
  - Basic e.g. Parallax IDE, MicroStudio

Communications

- Hardware
  - Hardwired
    - USB, Ethernet, RS-232, RS-485 network transceivers, CAN, I2C, SPI, microwire 1-wire
  - Wireless
    - Acoustic e.g. 38.5Khz Ultrasound transcievers
    - Optical e.g. 38.5KHz Infra-Red transcievers
    - Radio-Frequency e.g. 418MHz transceivers, BlueTooth, ZigBee, WiFi etc., GSM/GPRS modems
- Software
  - Client-server, master-slave configurations
- Terminal programs

System Development

- Strategies: for a software-only project the rule-of-thumb is:
  - 1/3 planning
  - 1/6 coding
  - ¼ component test and early system test
  - ¼ system test, all components in hand
- For an electronics project with embedded software (called firmware) you are working in two worlds and then connecting them!

Brooks, F., Jr. (1995) “The mythical man-month”, Addison Wesley

Practical electronics

- Start simple
- Prototype your ideas using ‘breadboard’ and terminal connector parts
- Build in a modular way separating functions in to self-contained units
- Test each module as you go along
- Test interfaces between modules
- What do to when it doesn’t work

Useful components to start

- Power source (e.g. an AC adapter)
- Prototyping ‘breadboard’, terminal blocks, hook-up wire
- Choose one of the self-contained embedded processor platforms e.g. Arduino, Propeller, Stamp, Javelin etc. as a prototyping base for your ideas
- Interface components to connect inputs to and output from you processor base e.g. pressure-pads, LEDs (Light-Emitting-Diodes)

Useful Tools

- Basic tools: small pliers, cutters, screwdrivers etc. can be obtained from DIY stores and shops like Maplin Electronics or on-line at http://rswww.com
- ‘Multi-meter’: a tool for measuring the properties of your circuit including current flow, resistance offered, potential difference, circuit continuity
- A PC, many code development and debugging tools run on them and you can ‘dry-run’ your ideas in simulators first

Where to find out more

- Urmenyi, B. (2001) ‘Electronics for Artists’, Bill Urmenyi Ltd http://www.urmenyi.co.uk
- Milford instruments http://www.milinst.com
- Parallax http://www.parallaxinc.com
- Maplin Website http://www.maplin.co.uk

- http://www.chrisleung.org/electronics.htm

Hazards

- Mains electricity can KILL
- Burns and fire hazards from soldering irons, exploding lead-acid batteries etc
- Eye-damage from laser radiation
- Some electronic components contain harmful chemicals see COSHH certificate
- Fumes generated from soldering may be harmful e.g. from lead core solder

Questions
Browse & demonstrations