Kinesthetics eXtreme - KX

Gail Kaiser

Columbia University

Kaiser@cs.columbia.edu

www.psl.cs.columbia.edu/dasada

George Heineman
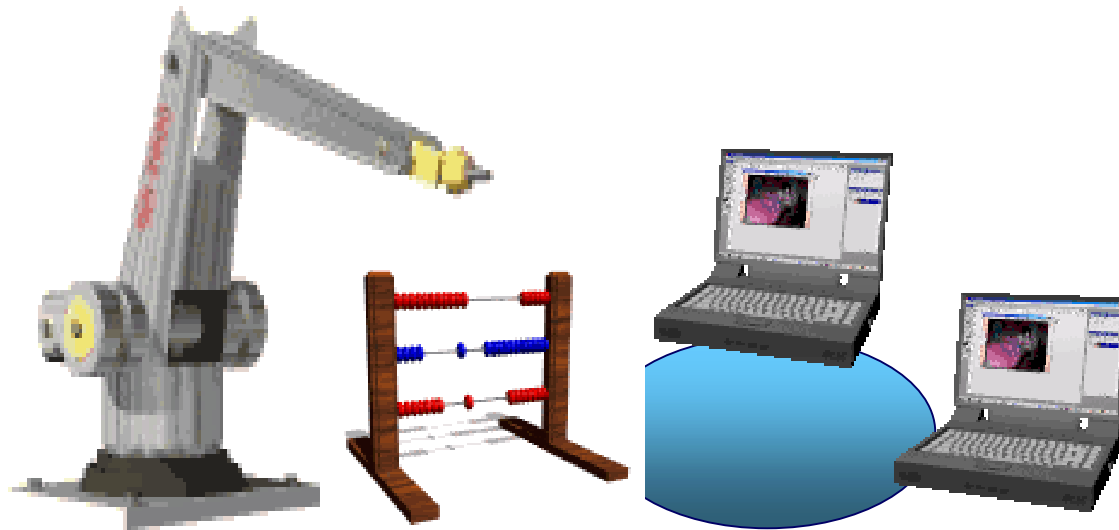
WPI

Heineman@cs.wpi.edu

www.cs.wpi.edu/~heineman/dasada
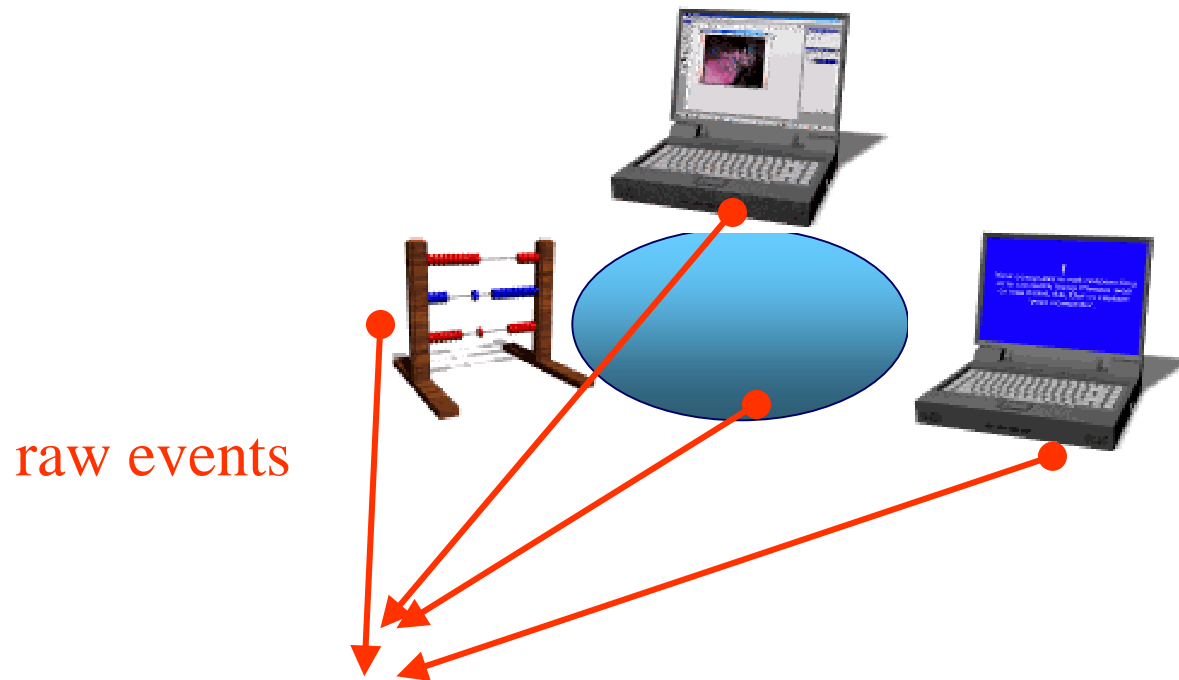
Scenario: Everything running smoothly

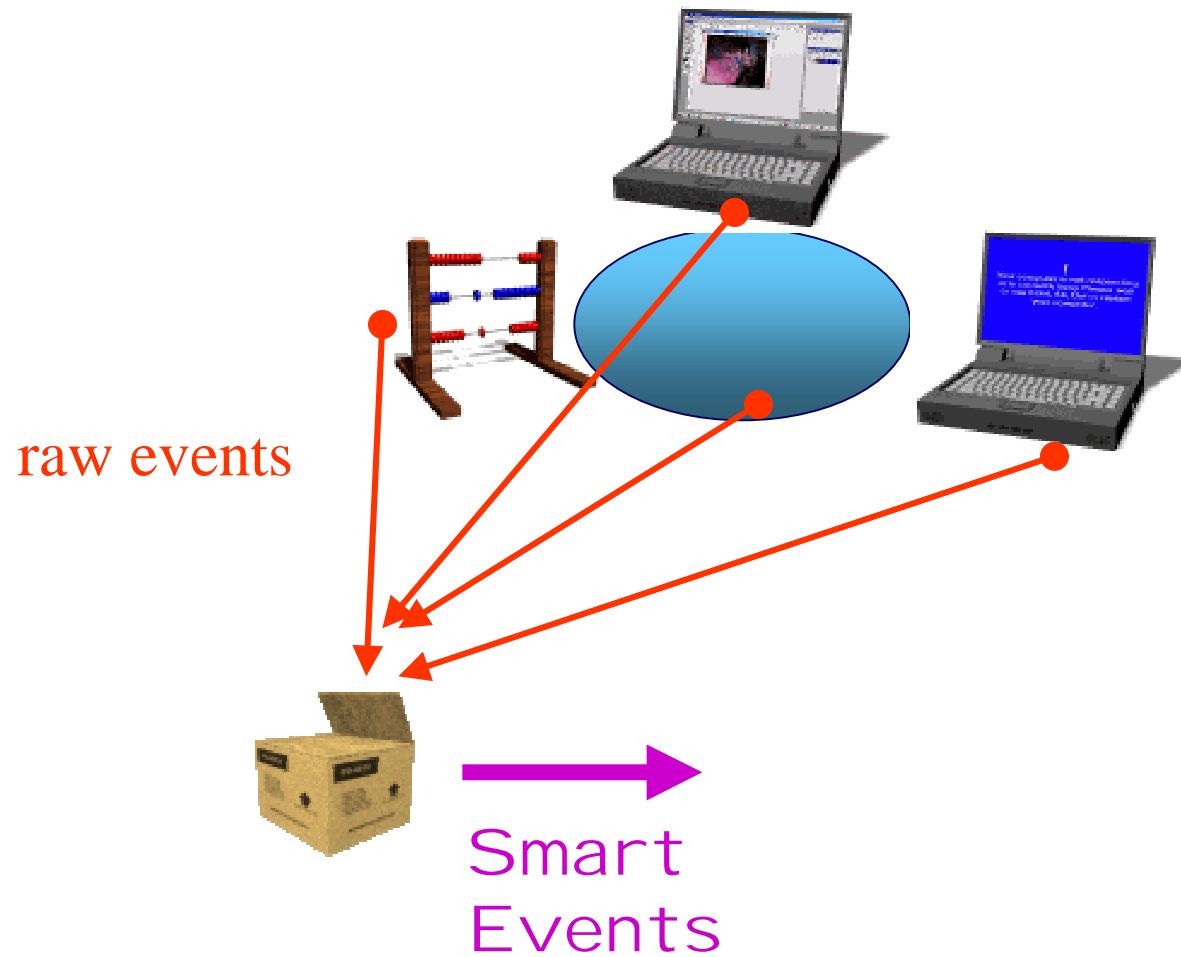# Component upgrade
(or environment change, or ...)

Oops

Let's probe into this

raw events

How is this done?

- **Active Interfaces** - callback model
- **AIDE** (Active Interface Development Environment) – auto-instrument code, incremental instrumentation updates/overlays architectural spec
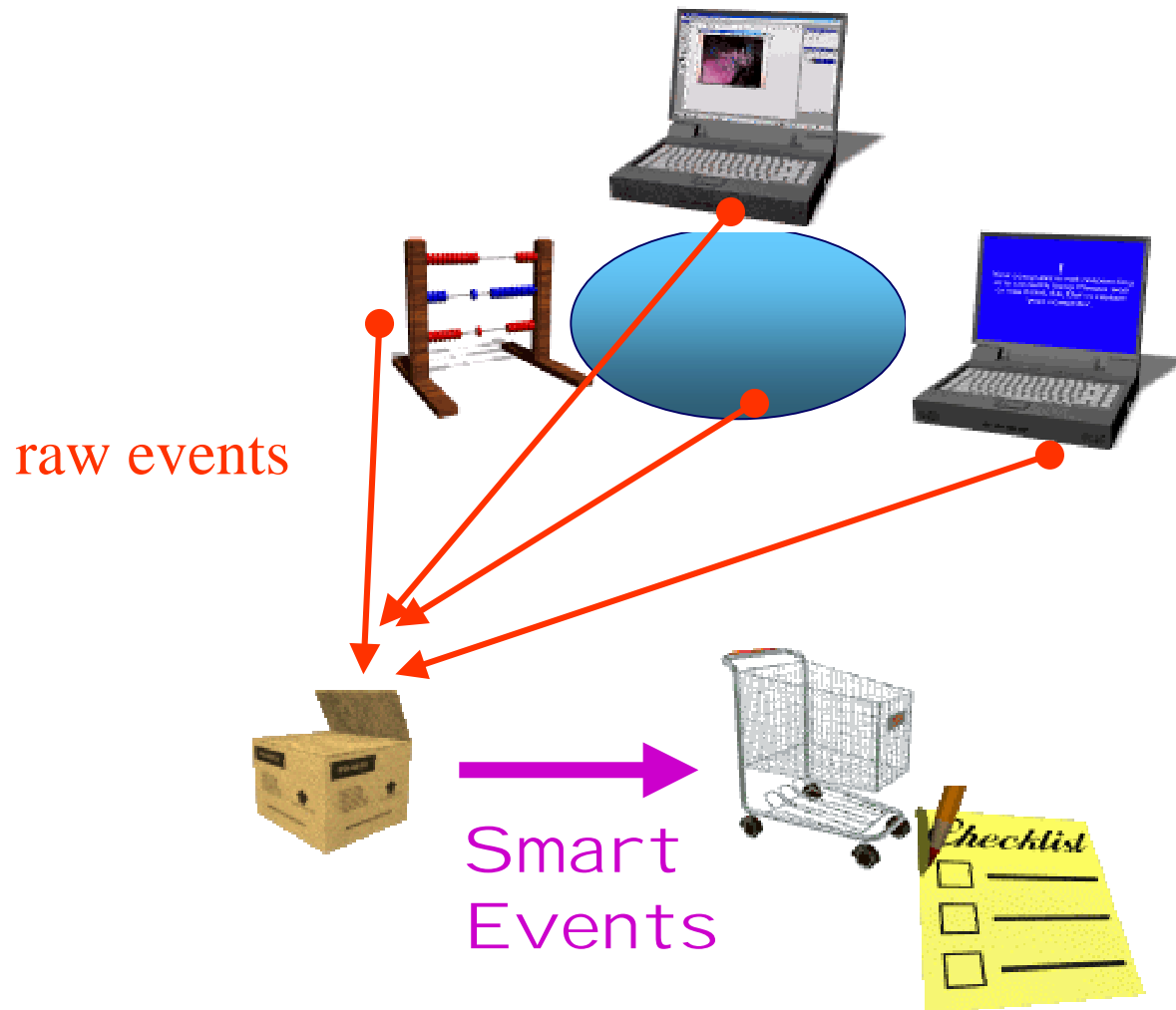- *Looking at ADLs, wrappers, interceptors*

Package things up

raw events

Smart
Events

How is this done?

- **Event Packager** – converts raw events to smart events via component-specific plugins (unless probes already convert)

- **FleXML** (Flexible XML) – enables smart events to "program" recognizers and gauges, supports dynamic introduction and composition of vocabularies and processors for event posets (prefixes), by bending some XML rules
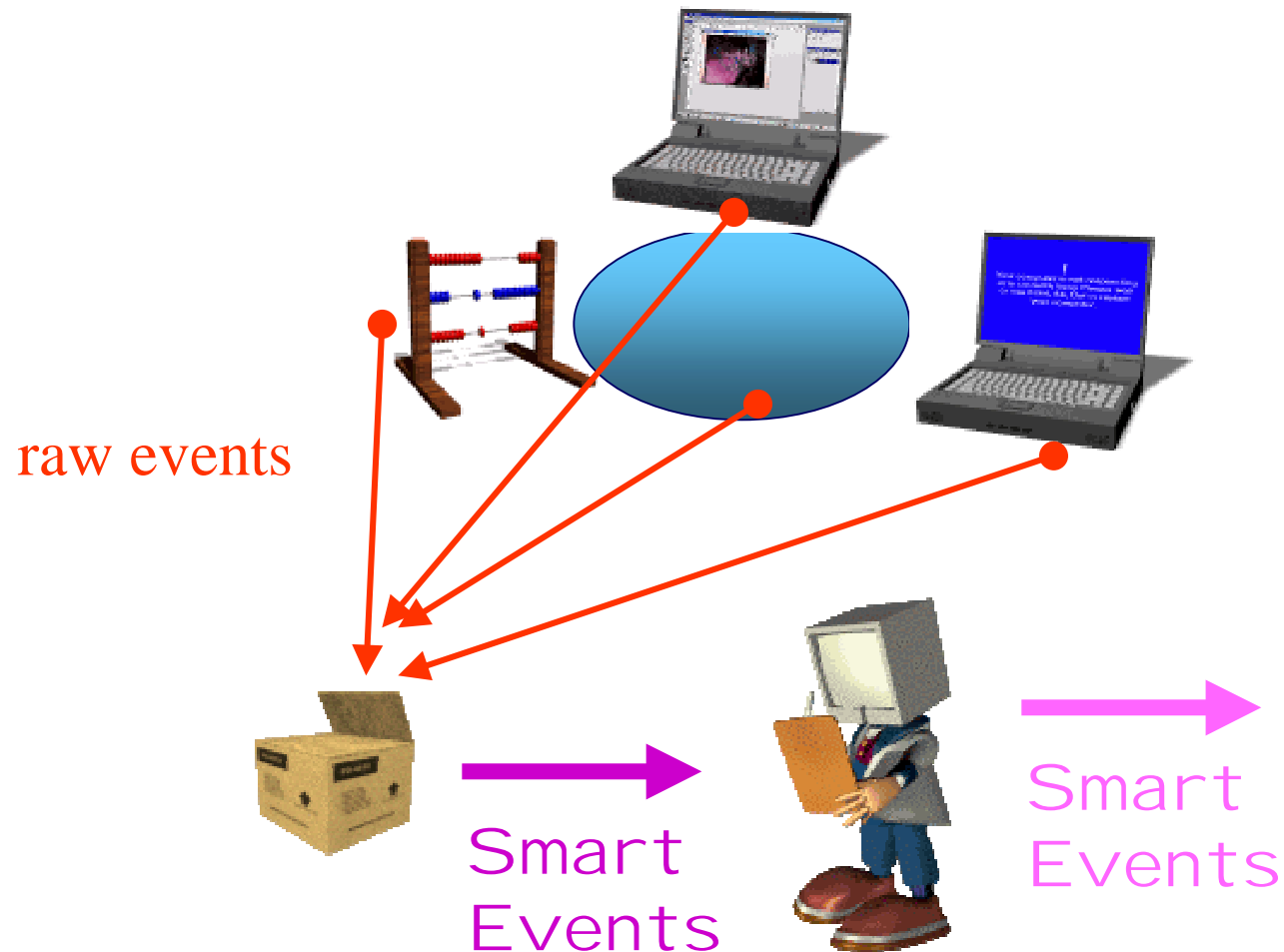
Check things out

raw events

Smart
Events

How is this done?

- **GC-4** (Groupspace Controller, to the four corners of the earth = version 4.x) – XML-based distributed event bus and service broker, mountable on other Internet event buses
- **Worklets** - mobile workflow agents install/update event patterns of interest, carry other mobile code payloads, perform other small tasks throughout KX
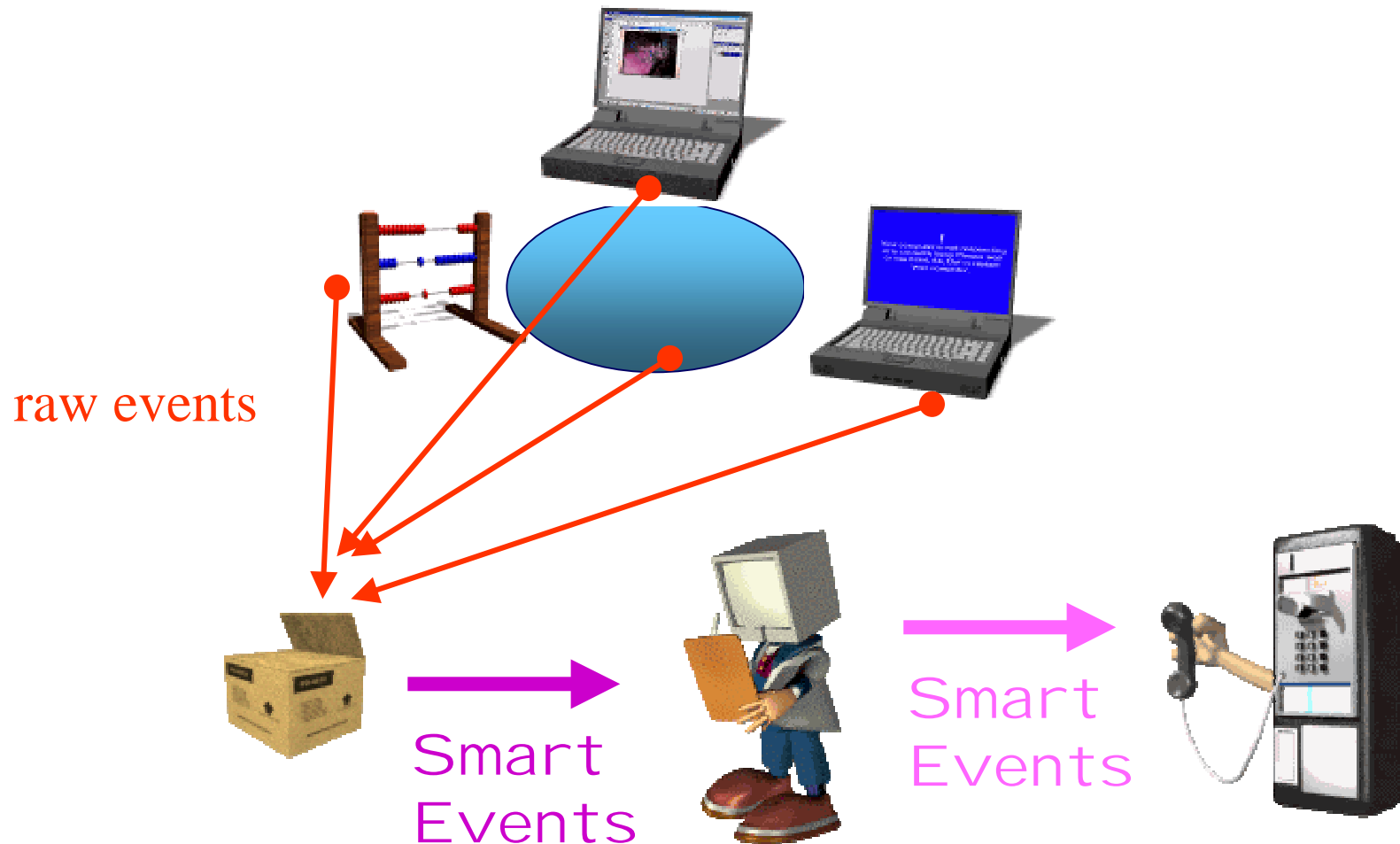- *Looking at complementary event bus and mobile code technologies*

Hmm... this doesn't look right

raw events

Smart
Events

Smart
Events

How is this done?

- **Event Distiller** – plugs in filters, complex event, state machine recognizers via FleXML
- **FleXML Meta-Parser** – validates against schema, invokes processors, consults Oracle for "unknown" tags, "invalid" stream prefixes
- **FleXML Oracle** – install new event vocabularies, recognizers, processors here
- *Looking at event stream notations, complex event recognizers, other pattern recognition*
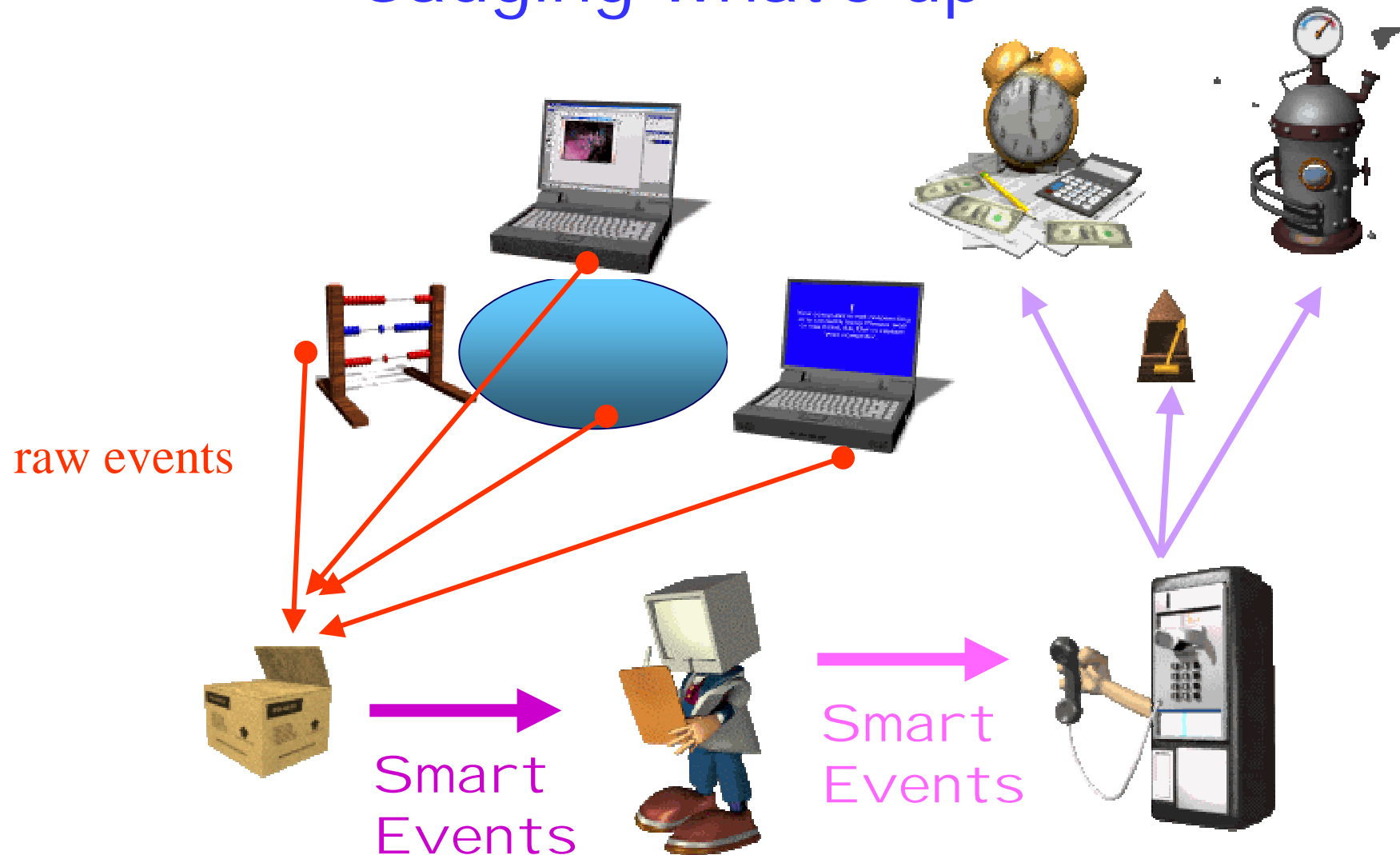- *Looking at XML Schemas, XSLT, XPath*

Get the word out

raw events

Smart
Events

Smart
Events

How is this done?

- **Event Notifier** – subscribes to (meta-) smart events from distillers, plugs in gauges via FleXML and worklets
- **Workgroup Cache** – caches FleXML schemas/processors, worklets, etc. throughout KX, "group"-based prefetch/push rules supports spectrum of lazy to eager deployment as well as semantics-based information sharing
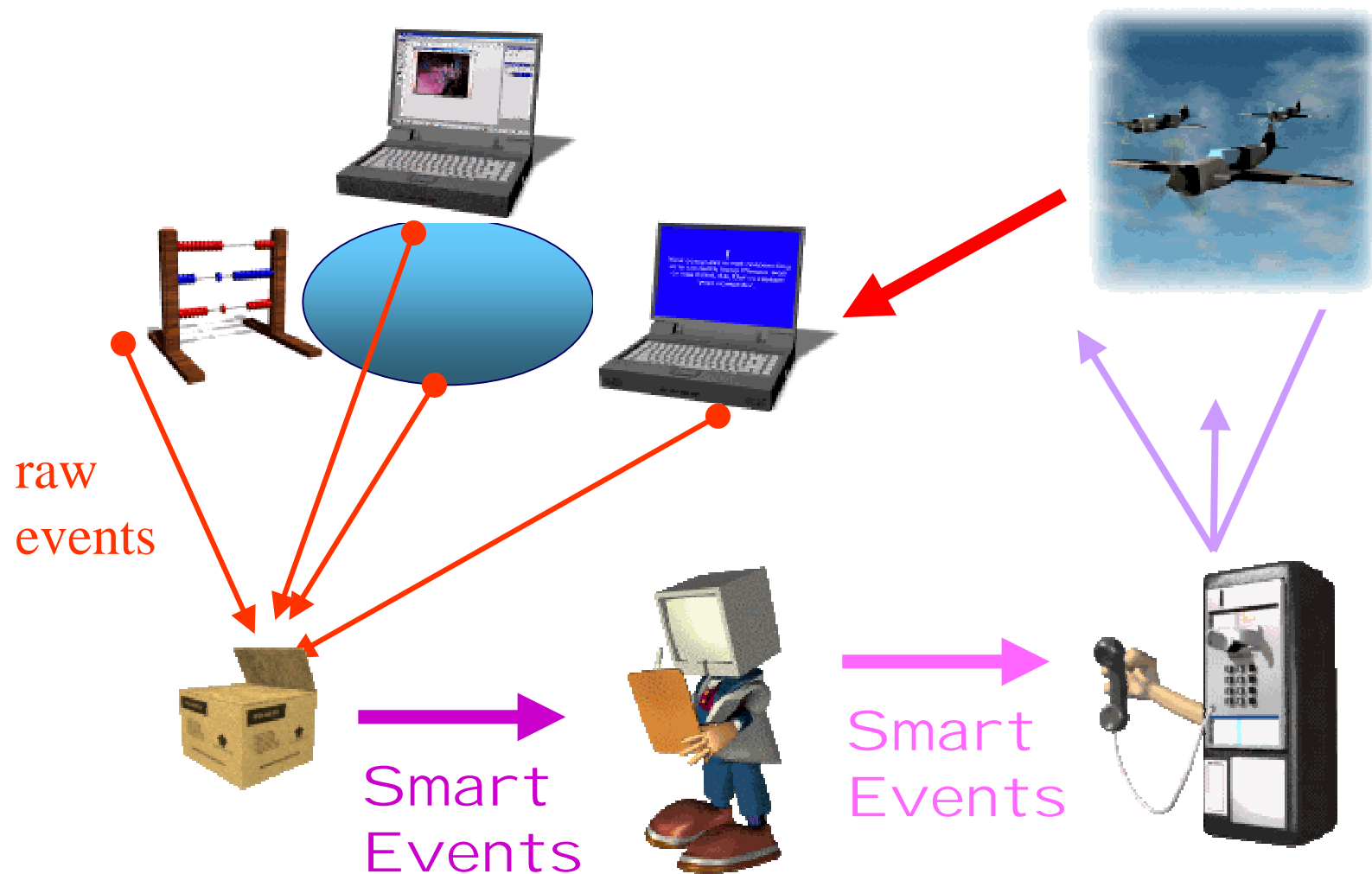
Gauging what's up

raw events

Smart Events

Smart Events

How is this done?

- **FleXML** processors - render smart events into visible gauges, reformat smart events for external analysis tools

- **TRIKX** (TRansitional Interface for KX) – portal organizes user or role-specific portlets for gauge presentation to human and/or automated decision agents, integrates GUI for FleXML and worklet development toolkits

- *Looking at "corporate portals", fisheye views, 3D immersion->2.5D flyover, XML browsers*
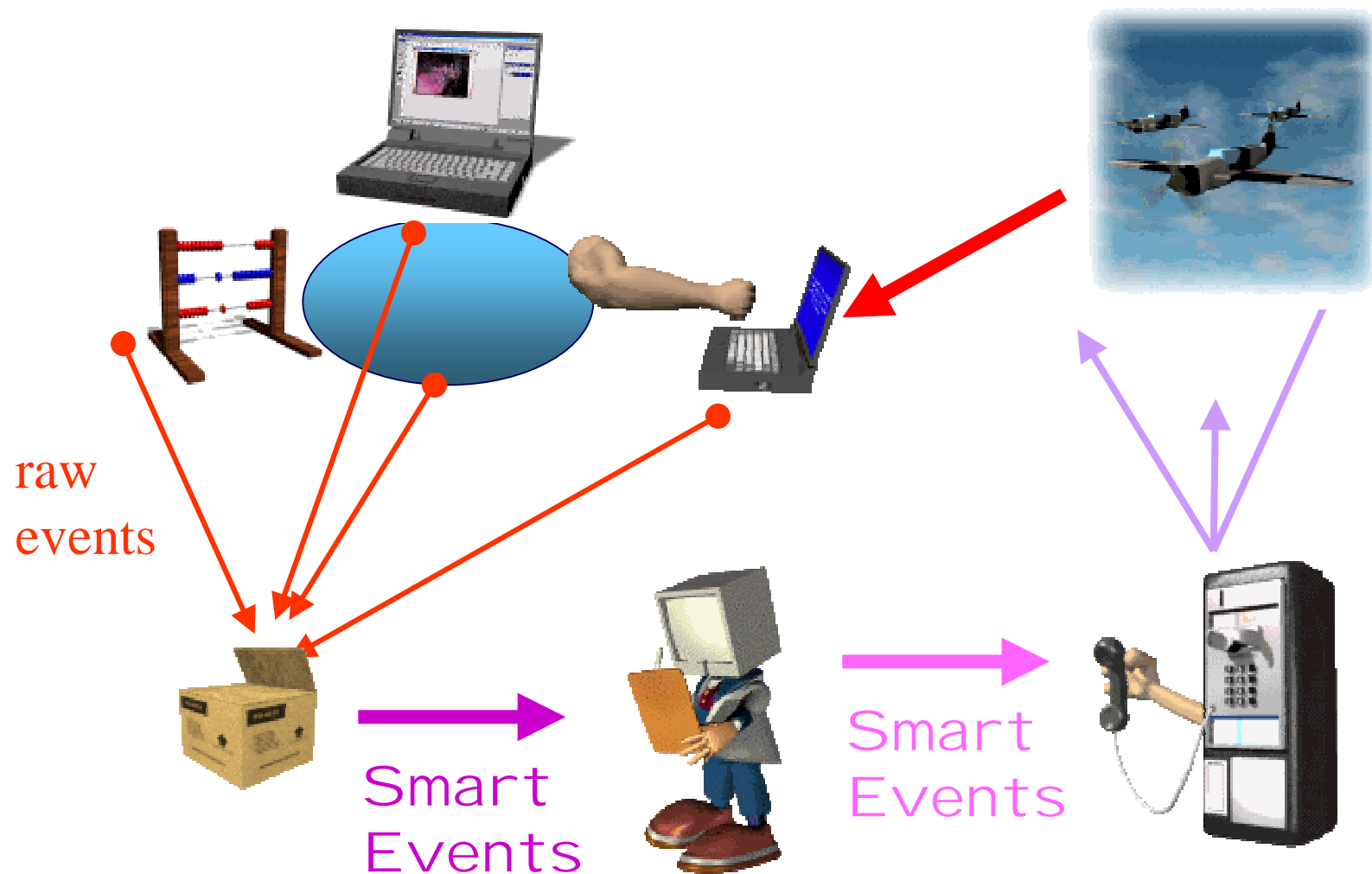
Send out the gaugents

raw
events

Smart
Events

Smart
Events

How is this done?

- **Gaugents** – specialized worklets oriented to reconfiguration, require Worklet Virtual Machine (WVM) adaptor for each target component/connector
- **Worklet junctions** - encapsulate mobile code (currently Java) for each stop along re-computable itinerary
- *Looking at run-time Java bytecode editing/modification/reinstantiation*
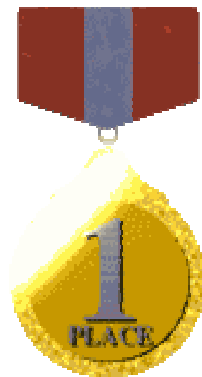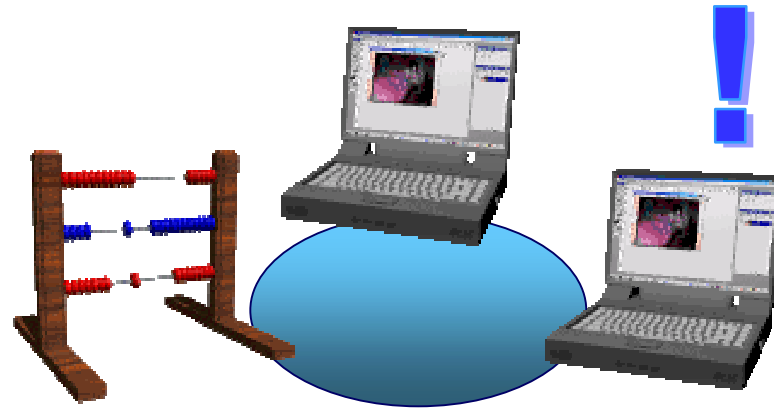
Reconfigure

raw
events

Smart
Events

Smart
Events

How is this done?

- **Process-aware systems** might coordinate reconfiguration (and KX monitoring), produce/consume worklets
- *Looking at decentralized process modeling, enactment, coordination*
- *Many other approaches...*

Yes!

What's available NOW? What's NEXT?

AIDE 1.0 (for Java)

http://www.cs.wpi.edu/
~heineman/dasada/

GC-2.5 (**GC-4** close)

Worklets 2.x
(moving target)

FleXML 0.9 close

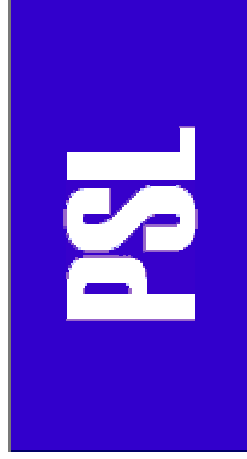http://www.psl.cs.columbia.
edu/software/download/

!DEMO!

XUES 0.5 (XML-based Universal Event Service = Event Packager, Event Distiller, Event Notifier) by December

Make up a Quad Chart

Ongoing collaborations wrt probe model, useful gauges, infrastructure architecture, event representation and recognition, decentralized process technologies, demo application

enable (vt) : *to make possible, practical, or easy*

PSL

PROGRAMMING SYSTEMS LAB
COLUMBIA UNIVERSITY

http://www.psl.cs.columbia.edu/

WPI WORCESTER POLYTECHNIC INSTITUTE