The Basics

The first usage would be to help make tangible the three ways feedback governs system behavior:

1. Amplifying feedback: feedback that accelerates system behavior, either in a positive or negative direction
2. Stabilizing feedback: feedback that keeps a dynamic system in homeostasis
3. Feedback Delays: the impact that latency has in influencing system dynamics

These three things are pretty generally accepted to define how any system is controlled. Some theorists use them as part of a catalog of system archetypes or patterns. There is, however, a theory of system control called cybernetics that uses a general framework of feedback to classify systems. I'm still getting my head around the details here, but the way I understand it is like this:

1. Open-Loop System: a system with a goal but no feedback. A simple lighting circuit is an example of this, where the goal is illumination, but the light created by the system does not influence the light created by the system.
2. First-Order System: a system that uses its output as input to reach its goal. A thermostat regulating room temperature is a classic example of this, where the thermostat turns on a heater that raises the temperature of the room until the thermostat senses that the temperature goal has been reached and so turns off the heater
3. Second-Order System: a system that uses its output as input to change its goal; a learning system. The thermostat first-order example becomes a second order system if one includes the human who, based on how warm or cold he feels, adjusts the thermostat in order to comfortable.

Notes based on Ashby

The goal of UX is to create a determinate machine: a given input produces the same output always

Operand = current affordance

Operator = user action (could be system action)

Transition = animation

Transform = resulting affordance

Transformation = transitions across an entire screen

Kinematic Graph = flow diagram

UX specification is complete if system is closed: all operands and transforms are defined

A machine state is an operand

Operands viewed through time are a trajectory = a workflow (a story board?)

Storyboards work to model animations because any continuous transition can be modeled as a series of discrete transitions

A UX specification aspires to be the canonical representation of the machine which embodies it

A transformation may be represented as a vector, the UX equivalent would be a model that shows all element changes screen to screen

Changing states is part of a transformation. Changing transformations is a mode.

A transient = overlay or interstitial

Coupling represents a connection between inputs and outputs. The coupler is the API.

Coupled machines form a new determined machine

Machines are coupled by specifying what will be the parameters of one machine (inputs) in terms of the other machines variables (outputs)

A machine is determined by its parts AND the couplings of its parts

An element has an immediate effect on another if it causes a change across a state transition

A diagram of immediate effects is a flow diagram or a control flow diagram

System complexity =

number of states,

number of variables,

number of components in a vector,

number of degrees of freedom

A complex system is made simpler by

statistics

assumptions

A persona is a subjective way of capturing statistical predictions