Autonomic Control of Composed Web Services

Dan Ionescu, Bogdan Solomon- Univ. Of Ottawa
Marin Litoiu- York University, Toronto
Gabriel Iszlai- IBM Toronto

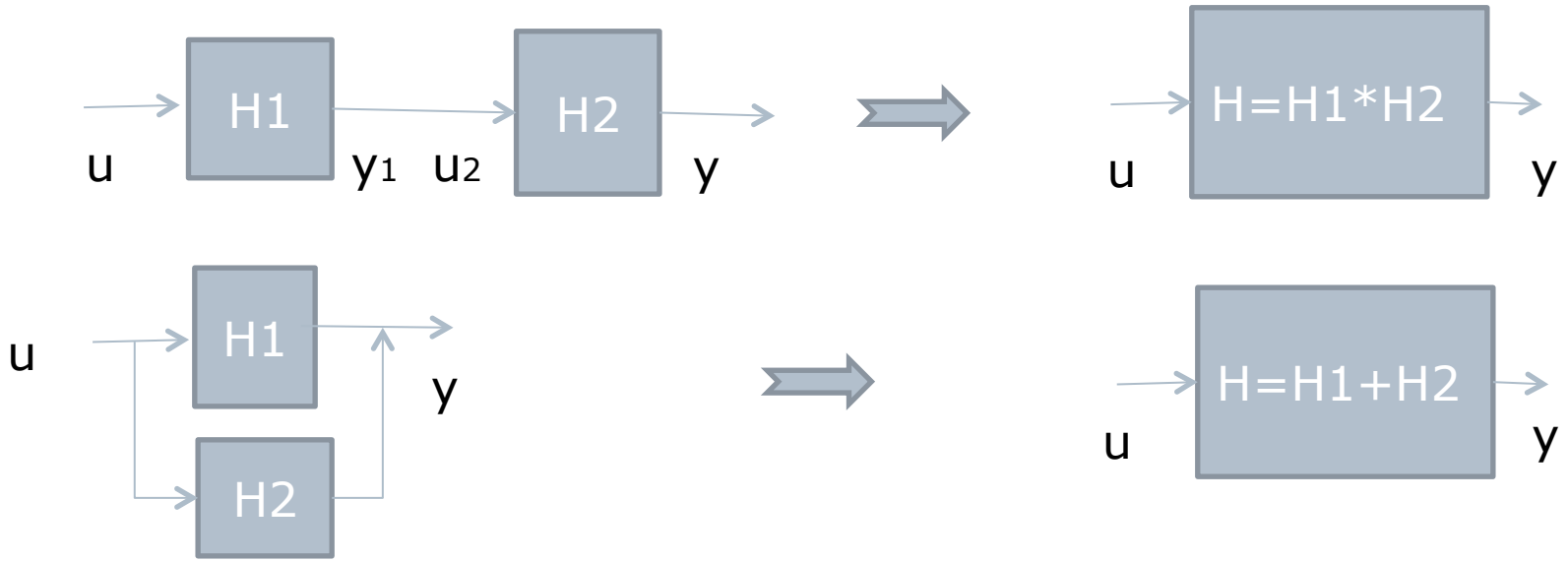
May 4, SEAMS 2010

Content

- Motivation
- Performance Models for WS
- Estimators
- Composite Models of WS
- Implementation
- Conclusions

Motivation(1)

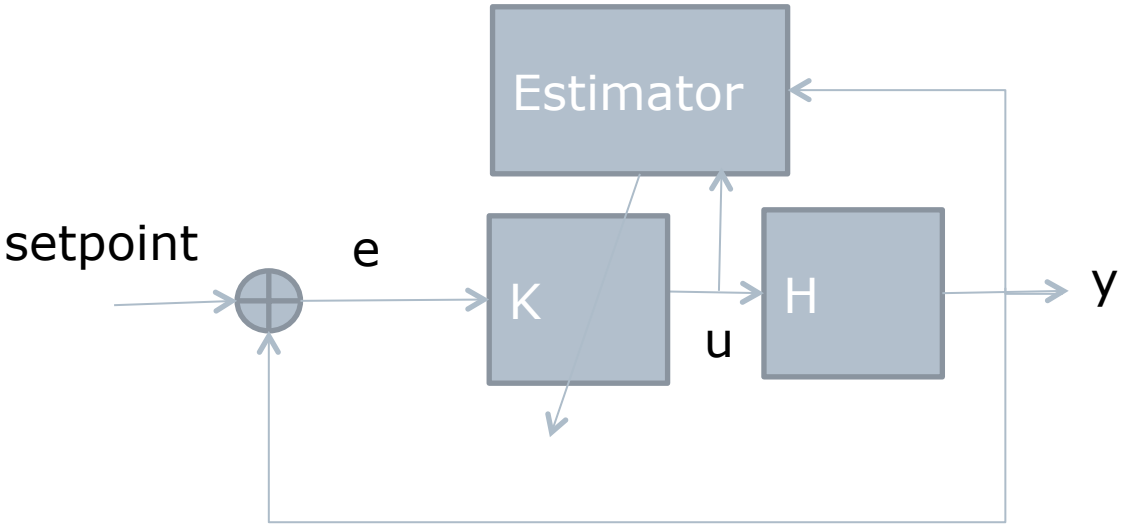
- In system theory, a system is described by a transfer function
- We can easily compose transfer functions, e.g.



- I can reduce any complex system to one transfer function H....

Motivation(2)

- If I have a system with a transfer function H, I know how to automate it....therefore I know to automate any system



- Estimator accounts for
  - Modeling errors
  - Measurement error
  - Time variant param.
- K: controller tr function

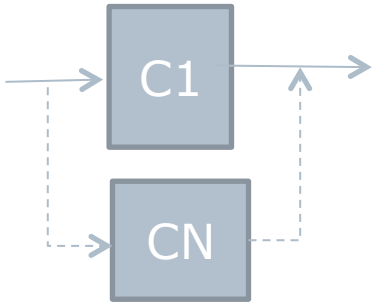
- Example

http://www.hrat.btinternet.co.uk/Homeostat.html

Motivation(3)

- Two mechanism of automatically scaling up and down in cloud is by:
- Multi-tier distribution of the load (series processing)

- Clustering (parallel processing)



- If I need to compose those patterns on line I need to know their combined effects

Motivation(4)

- Can we have the equivalent of transfer functions in software performance adaptation?
  - Estimate/predict quantitative dependencies inputs/ outputs
  - A systematic way of building controllers(autonomic managers)
- Can we compose those "transfer functions?"
  - Static and dynamic configuration of web services
- Can we have "Estimators" for software adaptation?
  - To deal with uncertainty

Performance Models for WS (non-linear)

Web Service (ES): states, inputs outputs

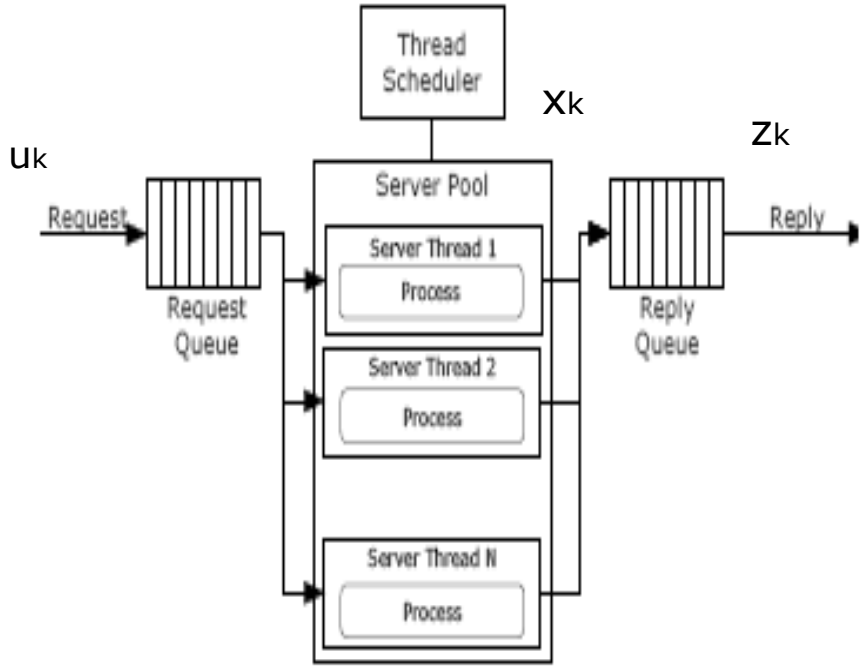
$$x_k = \begin{bmatrix} cpuLoad \\ responseTime \\ bufferSize \end{bmatrix}$$

$$u_k = [\ arrivalRate\ ]$$

.....

$$x_{k+1} = f(x_k, u_k)$$

$$z_k = h(x_k)$$



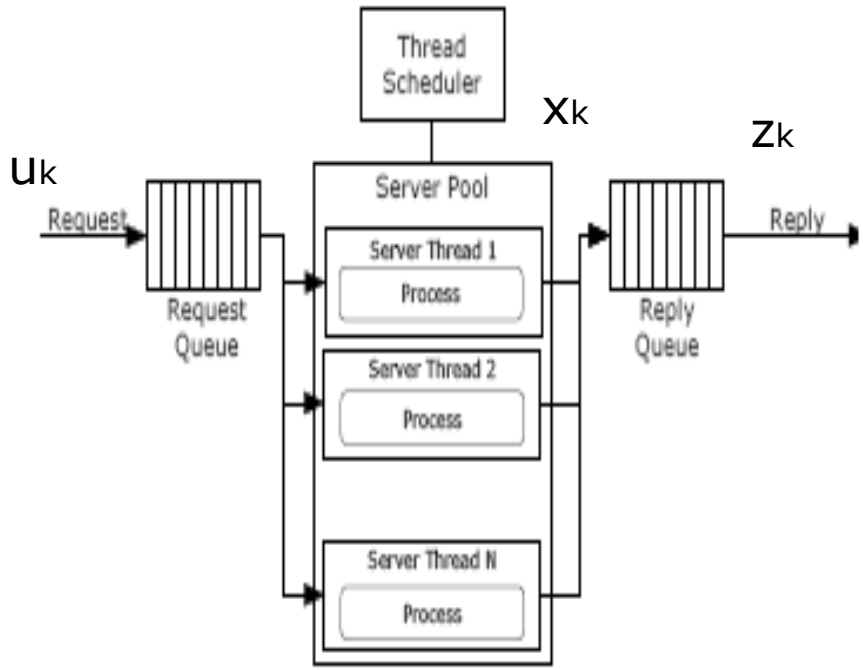
-k is the discrete time: k=0,1...
-x_k is the state, u_k is the input
-z_k is the measurement output at moment k, can be one of the x_k elements
-f and h are non linear functions....

Linear Performance Models (linear)

$$x_{k+1} = A_k x_k + Bu_k$$
$$z_{k+1} = H_{k+1} x_{k+1}$$

A, B, H are the Jacobians of f and h respectively, or..

•Partial derivatives around a functional point (x_k, u_k)



Linearized models are very important for "Estimators", see next slides

Estimators for WS

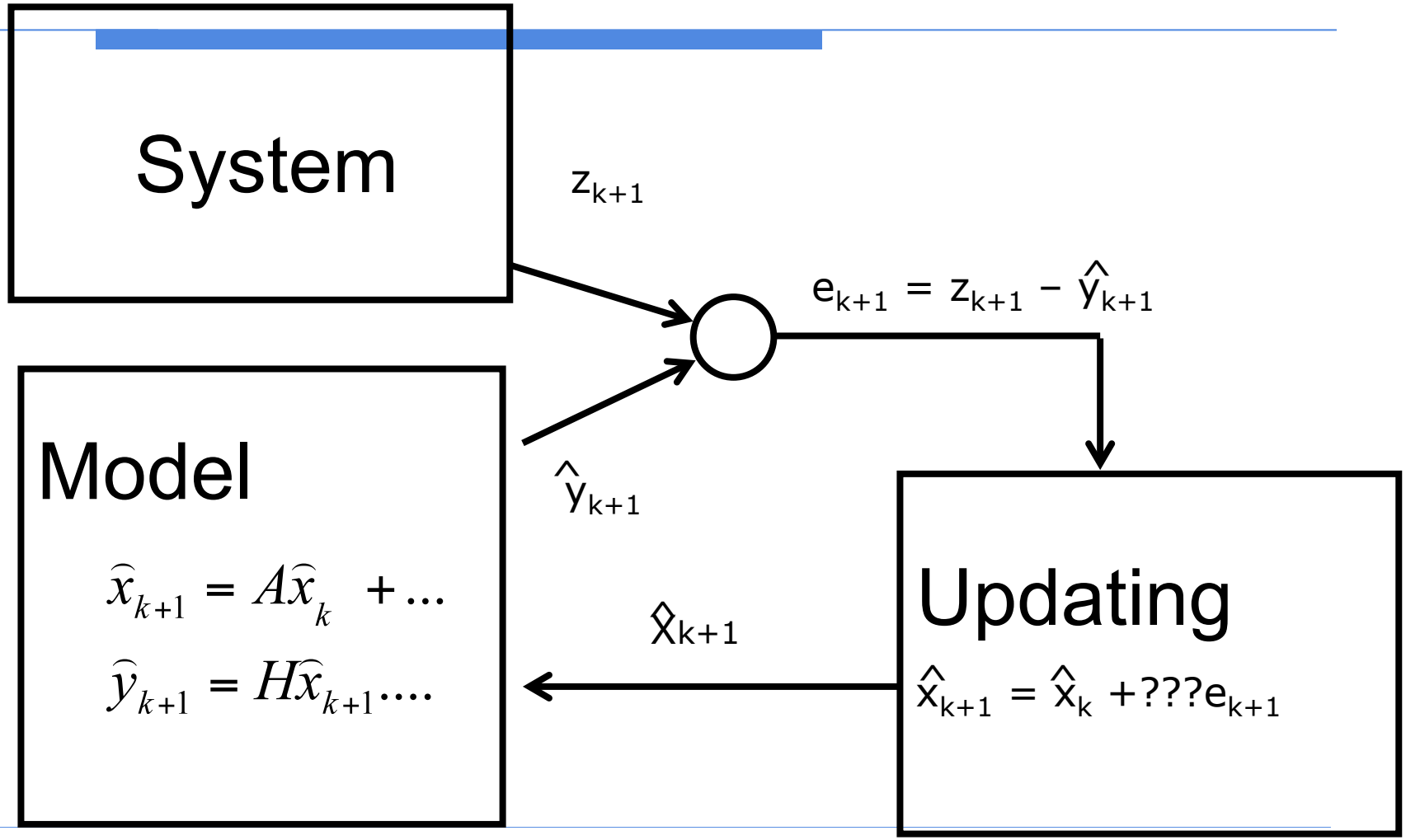
"All models are wrong, some models are useful."

- How do we deal with
  - Modeling errors
  - Measurement errors
  - Incomplete knowledge about the system
  - Time variant parameters
- Estimators make the best out of the above
- Initially applied to control systems (60's)
- now used in navigation
  - radar tracking of aircraft and ships
  - state = position, evolution is governed by dynamics
- and in control, e.g. adaptive robotics, signal processing, and other fields

The Kalman Filter for Linear Dynamic Systems

- The original filter was derived to give optimal estimates of time-varying states x_k:
  - Process model: $x_{k+1} = A_k x_k + Bu_k + \mathbf{w_k}$
  - Measurement model: $z_{k+1} = H_{k+1} x_{k+1} + \mathbf{v_{k+1}}$
  - w_k process noise, with the covariance matrix Q
  - v_k measurement noise, with the covariance matrix R
  - w_k and v_k- white, independent and with a normal distribution
- Minimize (in min mean square sense) both the prediction error (z_{k+1}-$H_k x_k$) and the parameter estimation error
  - conditional on:
    - the initial estimates of x_0
    - and P_0.…We define P_k = estimated covariance of estimates
    - and the observations z_i over 0 to k

The Extended Kalman Filter



Filter Equations for Linear Systems

- Predict $\mathbf{x}_{k+1}$ and observation $\mathbf{y}_{k+1}$:

$$\hat{\mathbf{x}}^{-}_{k+1} = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

$$\mathbf{y}_{k+1} = \mathbf{H}_{k+1} \mathbf{x}^{-}_{k+1}$$

- Predict the error covariance of $\hat{\mathbf{x}}^{-}_{k+1}$:

$$\mathbf{P}^{-}_{k+1} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T+\mathbf{Q}$$

- Kalman gain $\mathbf{K}$:

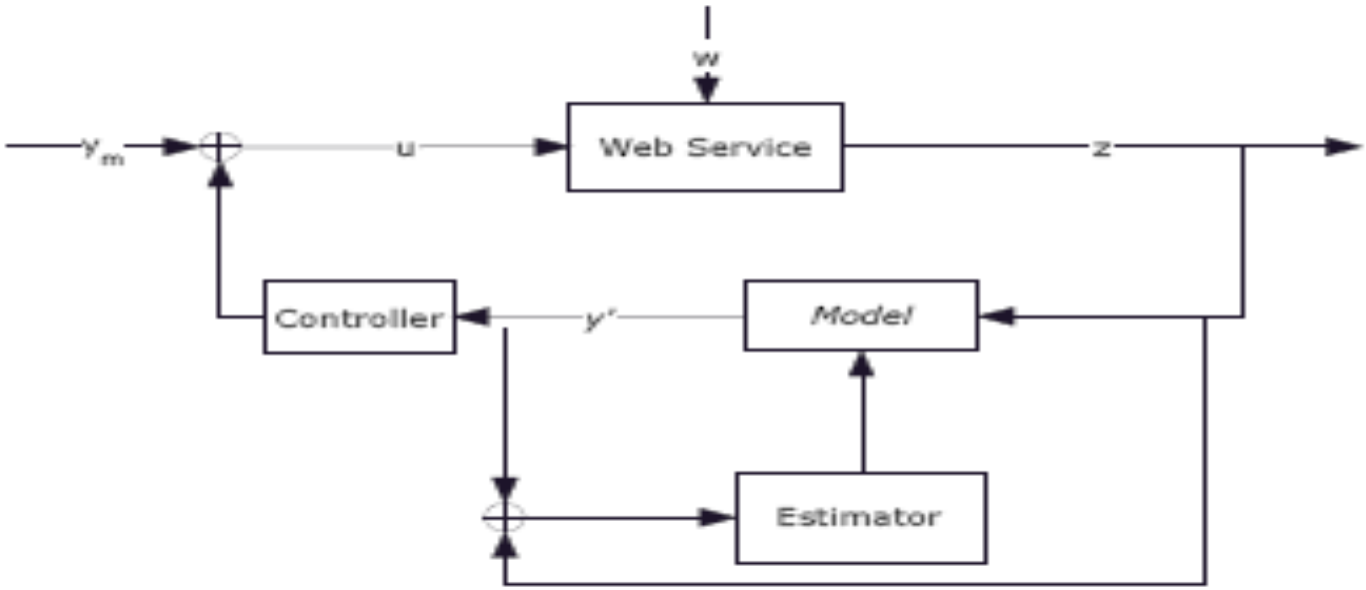
$$\mathbf{K}_k = \mathbf{P}^{-}_k\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^{-}_k\mathbf{H}_k^T + \mathbf{R})^{-1}$$

- Observe $\mathbf{z}_{k+1}$ and correct the estimate of $\mathbf{x}$:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_{k+1} - \mathbf{y}_{k+1})$$

- Update the error covariance $\mathbf{P}_k=(I-K_kH)\ \mathbf{P}^{-}_k$

Adaptive optimization



Controllers can be classic(synthesized from the model),
or optimization algorithms based when what if scenarios...

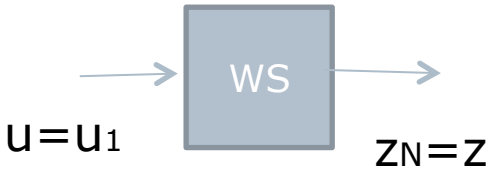
Composing: Series of WS

- Arrival rate define the input for the first service
- The output of each service (the throughput) is the input for the next service

$$x_N(k+1) = \begin{cases} f_N(x_N(k), u_N(k)), \text{for N=1} \\ f_N(x_N(k), z_{N-1}(k)), \text{for N>1} \end{cases}$$

$$z_N(k) = x_N(k)$$

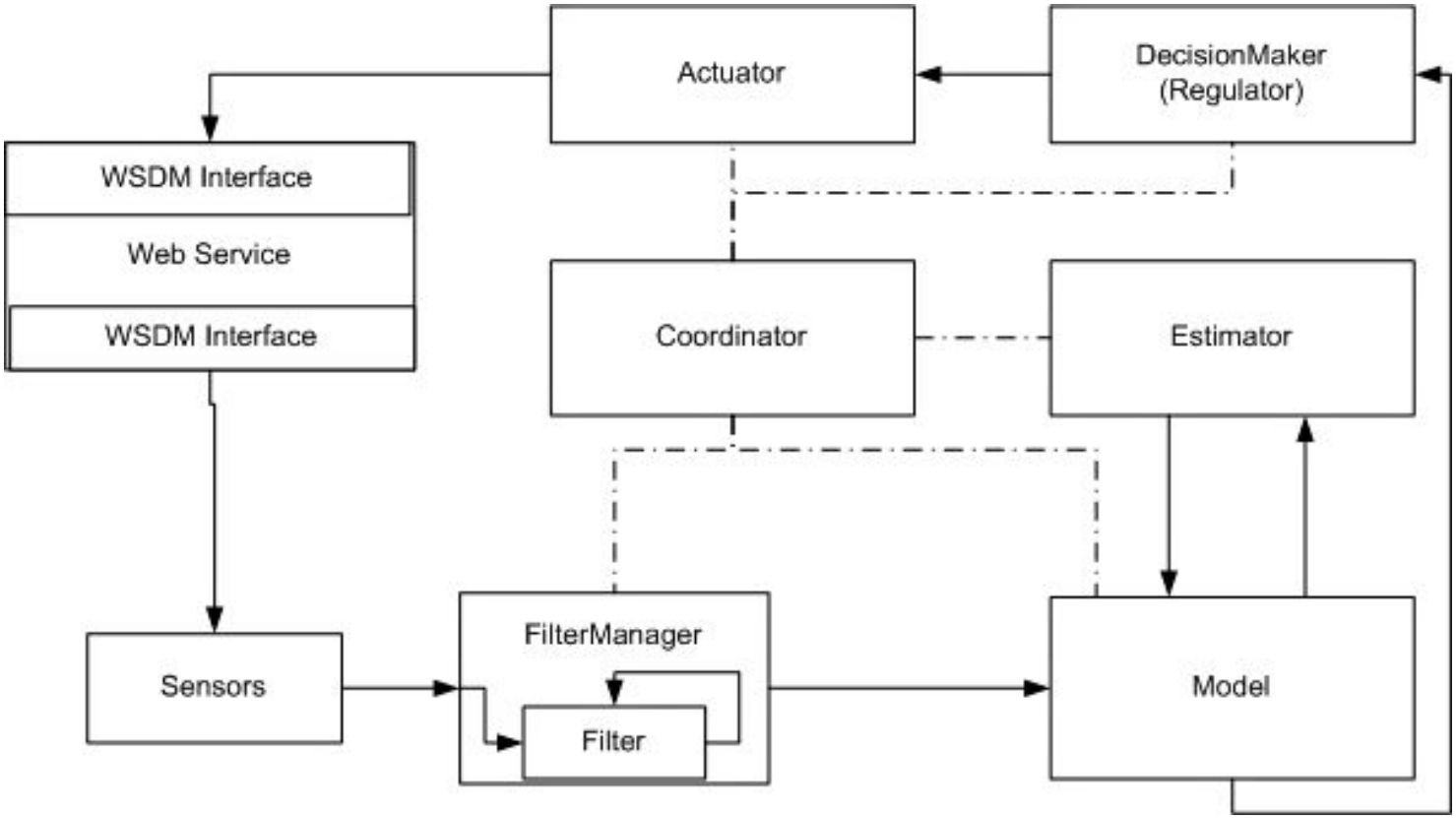
.....



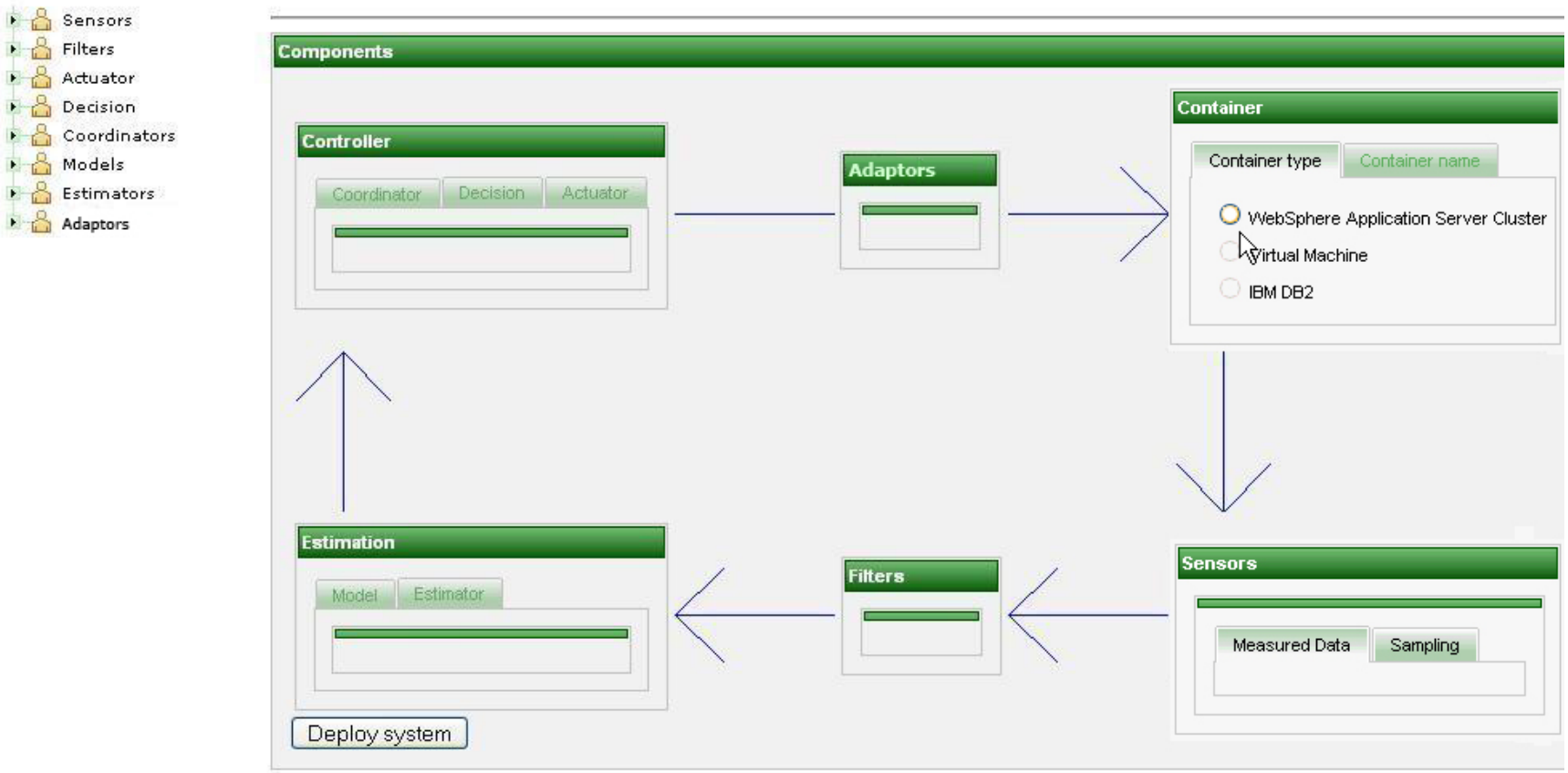
$$x_{k+1} = A_k x_k + Bu_k$$
$$z_{k+1} = H_{k+1} x_{k+1}$$

...now I can compute the Kalman estimator and the controller

Implementation

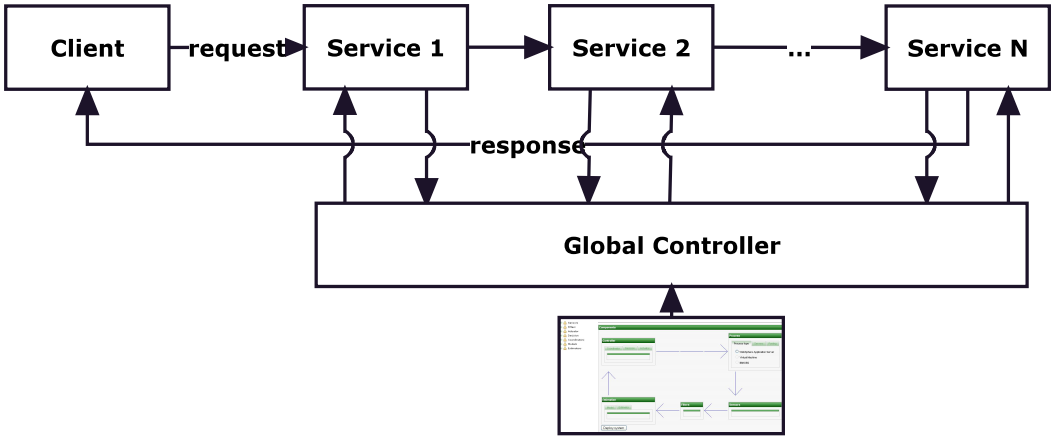


Implementation

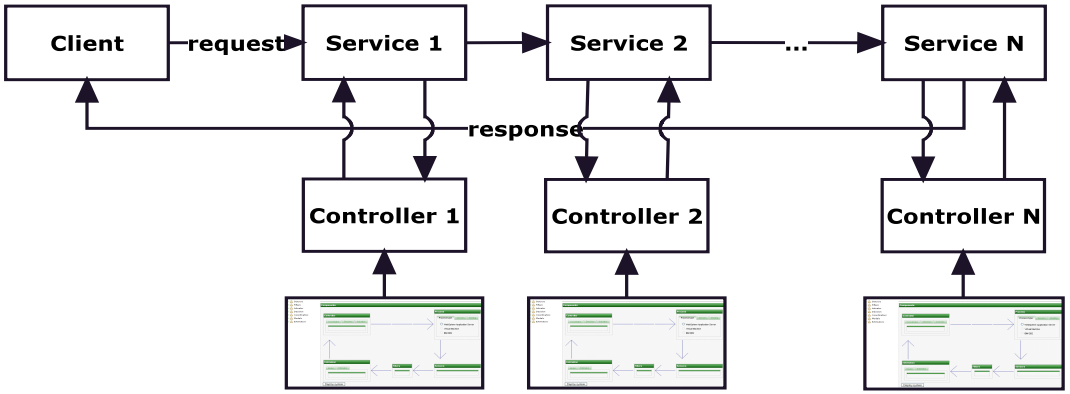


Serial Composition of WS

- Centralized control



- Decentralized control



Conclusions

- A performance model of WS can be expressed as an explicit non-linear function
- Linearization of the models allows the definition and implementation of estimators
  - Have two steps, prediction and correction
  - Kalman filter is an optimal estimator for linear systems
- With estimators and models with we can build adaptive performance control
  - The estimators compensate for modeling and measurement errors
- Composition of Web Services a needed approach for the automation of Web Services deployed On-the-Cloud