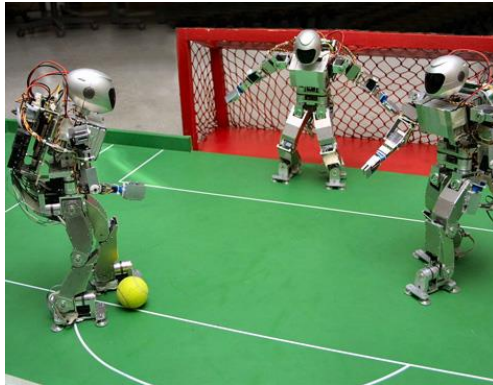


Ingeniería de Software II

Clase de C&C



Fútbol de robots

Ejercicio 7 de la práctica Tácticas de Arquitecturas

Enunciado

El presente ejercicio contempla el diseño de la arquitectura de un sistema de control de fútbol de robots. El propósito de este sistema es competir en un juego donde una serie robots físicos compiten por convertir la mayor cantidad de goles contra otro sistema idéntico a nivel físico (misma cantidad de piezas, mismo espacio de juego, misma configuración de cancha y pelota compartida)

El espacio de juego o cancha consiste en un marco de 200 cm x 100 cm, dos espacios de “arco” que son sensores de 40 cm dispuestos en cada lado de 100 cm de la cancha, y donde un objeto distinguido como la pelota se desliza a partir del impacto ejercido por los robots. Estos elementos pertenecen a la infraestructura común provista por la organización de la competencia.

Enunciado

Cada equipo provee para la competencia su propia infraestructura de juego consistente de 3 jugadores, una cámara de monitoreo para identificar el posicionamiento de los jugadores y de la pelota en la cancha, y un sistema que recibe las imágenes de esta cámara y toma decisiones para indicar sobre las movidas tácticas a los robots jugadores en pos de convertir la mayor cantidad de goles e imposibilitar que el equipo contrario convierta los propios. Cada uno de los 3 robots de cada equipo es distinguido con una cantidad de puntos en rojo o azul que distinguen por color a los equipos, y por la cantidad de puntos a un robot de otro dentro del mismo equipo para el sistema de control; la pelota es por otro lado coloreada con negro.

Un gol es convertido cuando uno de los 3 robots de algún equipo consigue que la pelota impactada “toque” uno de los arcos. Cada arco es asignado a uno de los equipos y el sistema que coordina el juego es responsable de controlar los tiempos de juego, y contar la cantidad de goles que cada uno de los equipos va convirtiendo.

Enunciado

La cámara común genera 3 imágenes por segundo, las cuales son entregadas en forma simultánea a cada sistema de control de robots quienes las someten a dos acciones: identificar la posición de la pelota e identificar la posición de todos los jugadores (propios y contrarios).

La identificación de la posición de la pelota se basa en aplicar un algoritmo de filtro que detecta el negro de la pelota por encima de los otros colores en forma eficiente. Esta identificación es optimizada al reducir la porción de la imagen a procesar al mínimo posible en función del conocimiento de la posición de la pelota que se tiene en base al procesamiento previo. Gracias a esta optimización es posible computar una porción de imagen por señal recibida (> 3 computaciones por segundo de filtro). Pero, si con el algoritmo optimizado no se encontrase la pelota en el espacio buscado, se procede a buscar sobre toda la imagen de cancha en pos de localizar la pelota, lo cual demora aproximadamente 1 segundo (3 señales de cámara).

Enunciado

La identificación de los robots se realiza a partir de aplicar un filtro de rojos y azules (dos filtros) demorando cada uno 1 segundo. Esto hace que se pueda identificar la posición de los jugadores cada 6 muestreos de la cámara. El movimiento de los jugadores se realiza a partir de dos actuadores, uno de dirección y otro de velocidad.

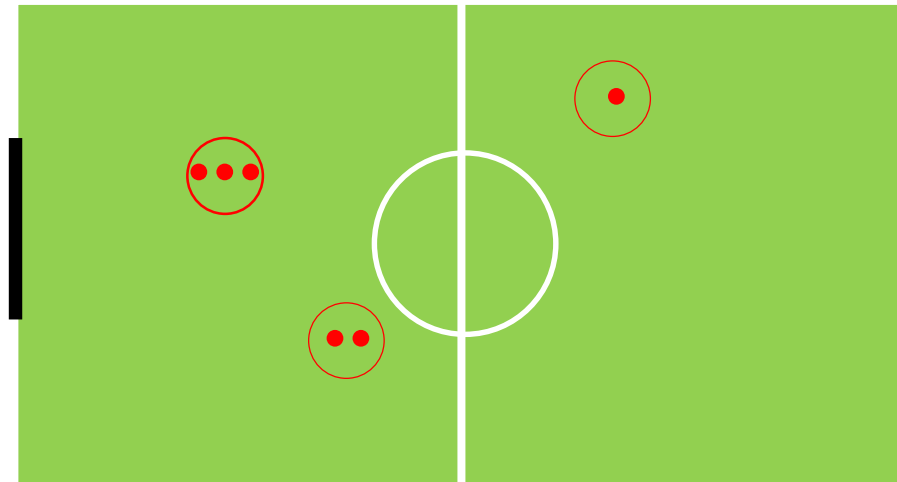
Se requiere que el sistema que controla la estrategia del juego de cada equipo guíe a los robots a convertir la mayor cantidad de goles posibles y evitar que el contrario haga los propios, guiando a los robots a través de sus actuadores. Dado que constantemente se encuentran mejoras en las estrategias, el sistema debe ser flexible a la hora de cambiar de una estrategia a otra. En ciertas ocasiones, las imágenes pueden venir con defectos que dificultan la identificación, con lo cual los filtros informan datos erróneos. El sistema debe en estos casos descartarlos y utilizar la historia y proyecciones de los movimientos.

a) Identifique los subsistemas del dominio, identifique y especifique los 3 escenarios más relevantes de acuerdo a su criterio, determine qué tácticas usa y especifique el modelo de C&C asociado con el sistema descrito.

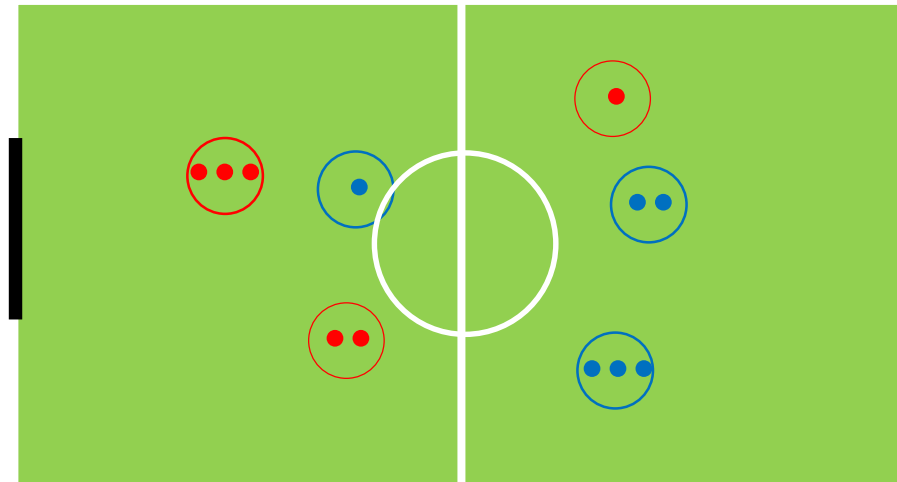
Entendiendo el Problema...



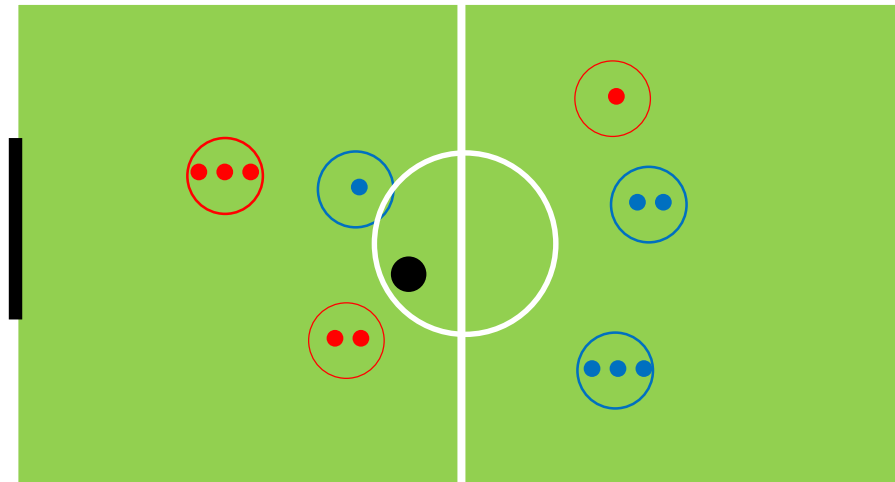
Entendiendo el Problema...



Entendiendo el Problema...



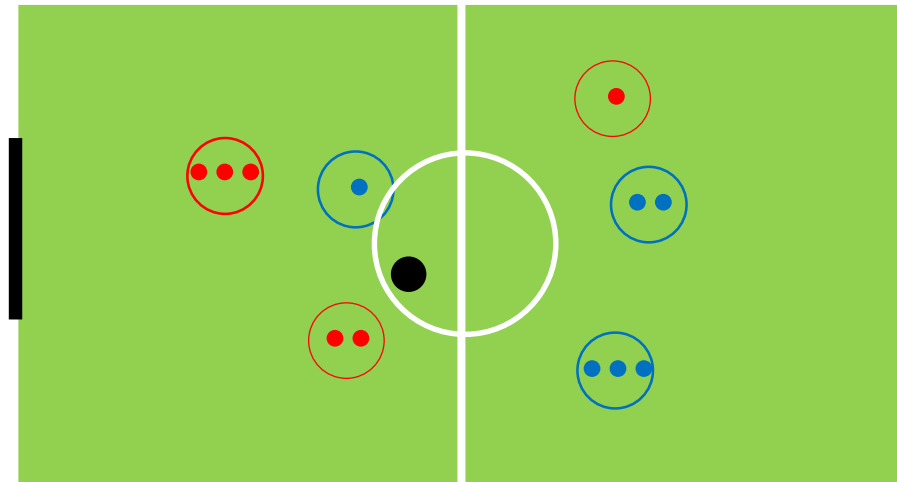
Entendiendo el Problema...



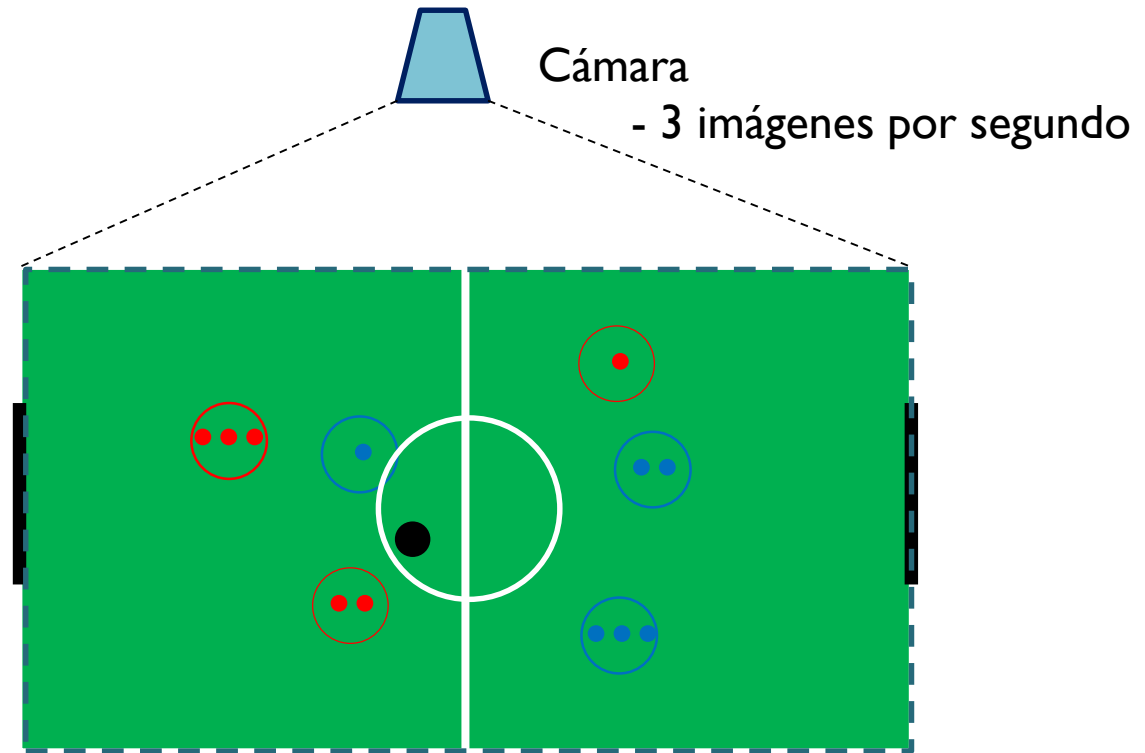
Entendiendo el Problema...



Cámara



Entendiendo el Problema...



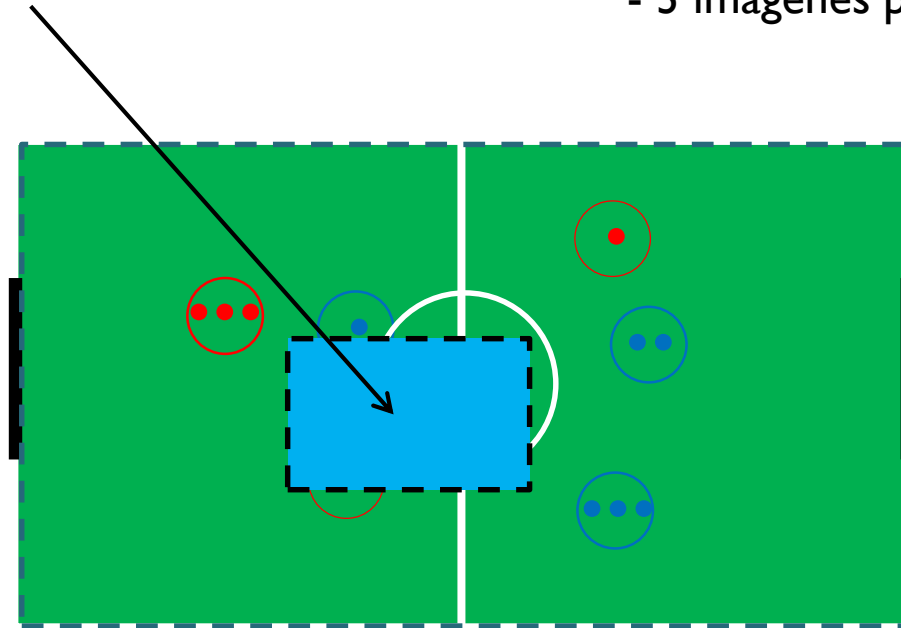
Entendiendo el Problema...

Filtros para la pelota



Cámara

- 3 imágenes por segundo



Entendiendo el Problema...

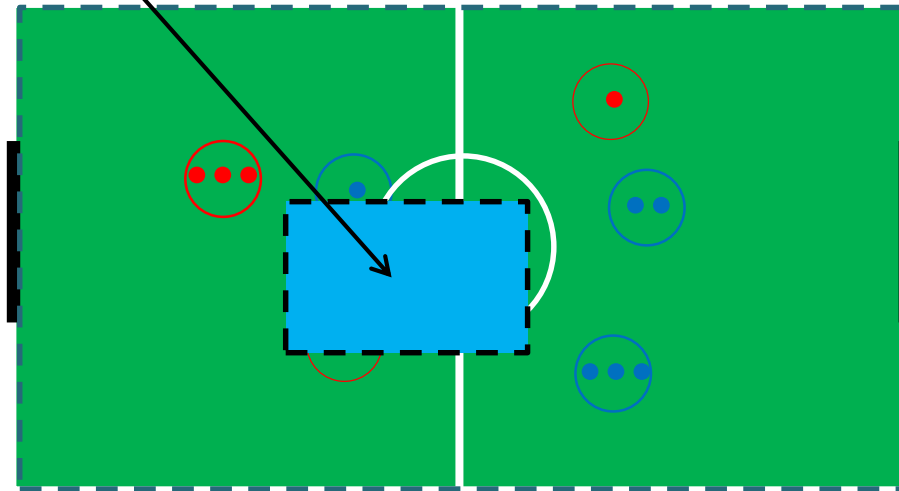
Filtros para la pelota

- Optimizado
*una porción de imagen
por señal recibida*



Cámara

- 3 imágenes por segundo



Entendiendo el Problema...

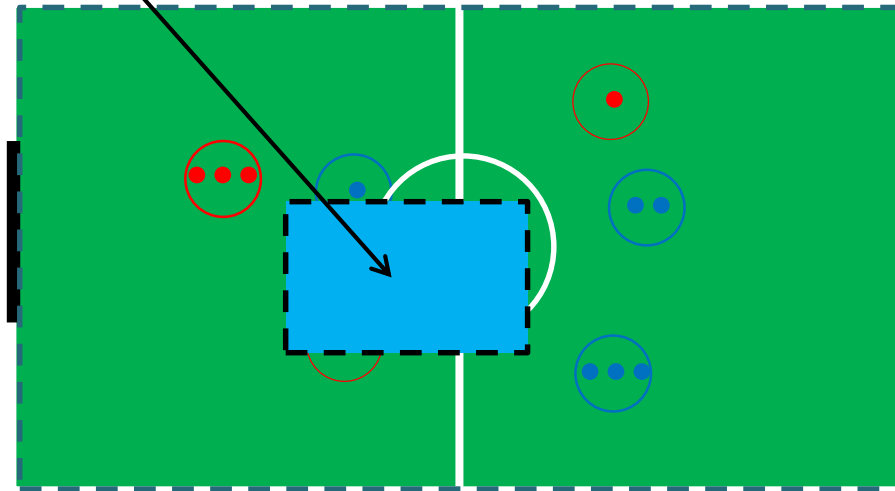
Filtros para la pelota

- Optimizado
 > 3 *muestreos x seg.*



Cámara

- 3 imágenes por segundo



Entendiendo el Problema...

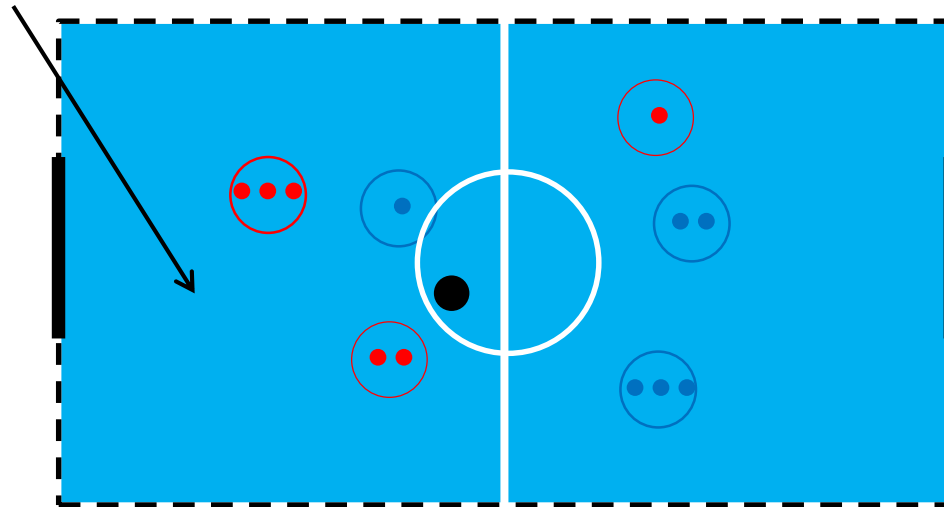
Filtros para la pelota

- Optimizado
 > 3 muestreos x seg.
- Total
 demora aprox.
 1 segundo



Cámara

- 3 imágenes por segundo



Entendiendo el Problema...

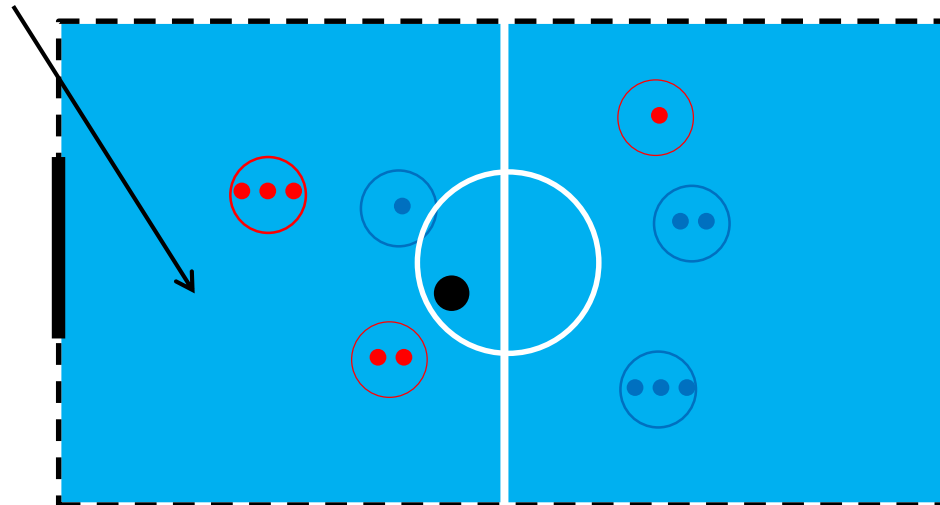
Filtros para la pelota

- Optimizado
 > 3 muestreos x seg.
- Total
 1 muestreo x seg.



Cámara

- 3 imágenes por segundo



Entendiendo el Problema...

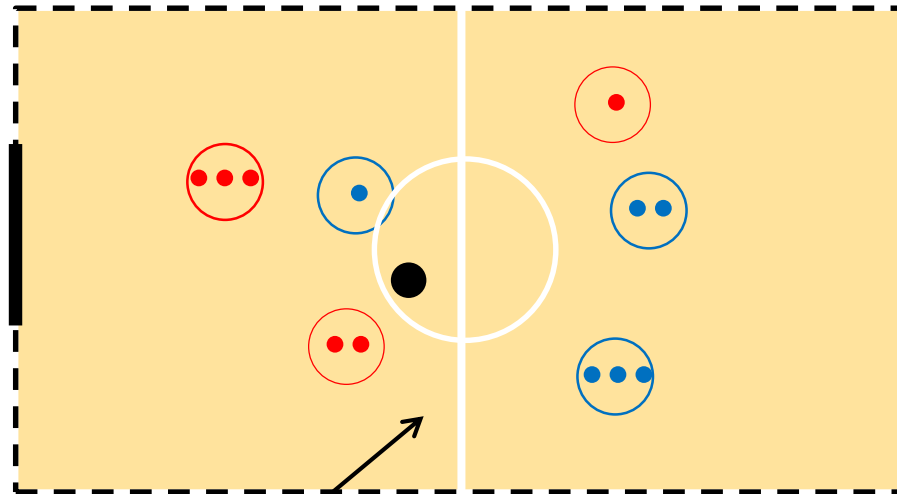
Filtros para la pelota

- Optimizado
 > 3 muestreos x seg.
- Total
 1 muestreo x seg.



Cámara

- 3 imágenes por segundo



Filtros para los jugadores

- Equipo rojo
 1 muestreo x seg.
- Equipo azul
 1 muestreo x seg.

Entendiendo el Problema...

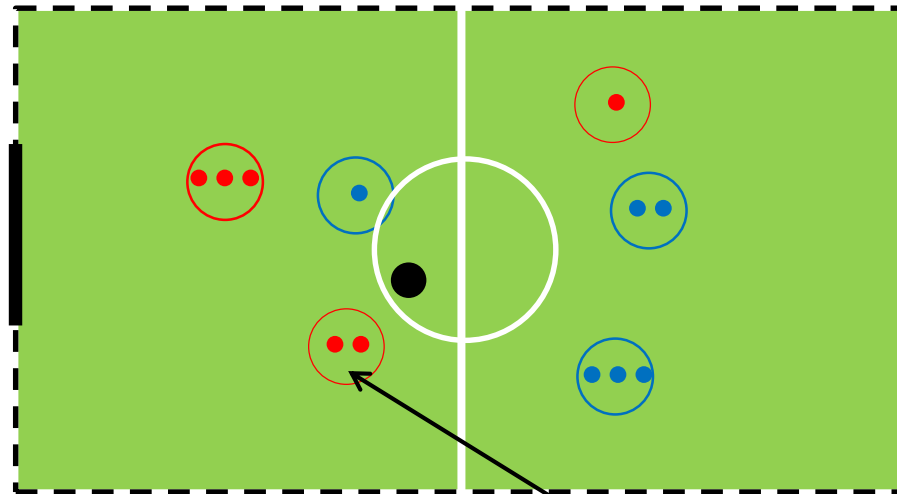
Filtros para la pelota

- Optimizado
 > 3 muestreos x seg.
- Total
 1 muestreo x seg.



Cámara

- 3 imágenes por segundo



Filtros para los jugadores

- Equipo rojo
 1 muestreo x seg.
- Equipo azul
 1 muestreo x seg.

Control de los robots

- Actuador Dirección
- Actuador Velocidad

Resumiendo...

Cámara

- 3 imágenes por segundo

Filtros para la pelota

- Optimizado
 $> 3 \text{ muestreos } \times \text{ seg.}$
- Total
 $1 \text{ muestreo } \times \text{ seg.}$

Filtros para los jugadores

- Equipo rojo
 $1 \text{ muestreo } \times \text{ seg.}$
- Equipo azul
 $1 \text{ muestreo } \times \text{ seg.}$

Control de los robots

- Actuador Dirección
- Actuador Velocidad

Resumiendo...

Cámara

- 3 imágenes por segundo

Filtros para la pelota

- Optimizado
 $> 3 \text{ muestreos } \times \text{ seg.}$
- Total
 $1 \text{ muestreo } \times \text{ seg.}$

Filtros para los jugadores

- Equipo rojo
 $1 \text{ muestreo } \times \text{ seg.}$
- Equipo azul
 $1 \text{ muestreo } \times \text{ seg.}$

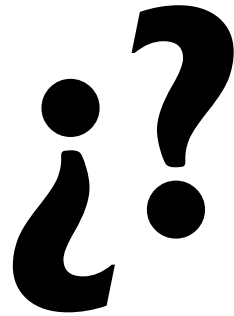
Control de los robots

- Actuador Dirección
- Actuador Velocidad

¿Qué tenemos que resolver?

- a) Identificar los subsistemas del dominio
- b) Identificar y especificar los escenarios
- c) Especificar el modelo de C&C asociado con el sistema descrito.
- d) Enumerar tácticas utilizadas

Identificar los subsistemas del dominio...



Identificar los subsistemas del dominio...

Filtrador Imágenes

- Encargado de procesar las imágenes que llegan desde la cámara

Actuador

- Encargado de controlar los robots

Subsistema de Estrategia

- Encargado de determinar los movimientos de acuerdo a las posiciones de los robots y la pelota

Identificar los escenarios...



Identificar los escenarios...

-Escenario #1

-Performance:

- Identificar posición pelota y jugadores en menos de 1 segundo

Fuente	• Cámara
Estímulo	• La cámara envía una imagen
Artefacto	• Filtrador de Imágenes
Ambiente	• Funcionamiento Normal
Respuesta	• Se identifica la posición de la pelota y los jugadores
Medida de la respuesta	• Máximo 1 segundo

Identificar los escenarios...

-*Escenario II*

-<Categoría> (?): *Detección de datos outliers*

Fuente	• Interno al sistema
Estímulo	• Se calculó la posición de los jugadores y la pelota
Artefacto	• Subsistema de estrategia
Ambiente	• Los datos calculados tienen errores
Respuesta	• Se descartan los outliers y se utilizan estimaciones
Medida de la respuesta	• El 90% de los casos se descartan los outliers correctamente

Identificar los escenarios...

-Escenario #3

-Modificabilidad: *Cambiar la estrategia para el juego*

Fuente	• Equipo de desarrollo
Estímulo	• Se desarrolló una nueva estrategia para el juego
Artefacto	• Sistema
Ambiente	• En tiempo de desarrollo
Respuesta	• Se modifica la estrategia a utilizar
Medida de la respuesta	• No se requirieron modificaciones al código del sistema

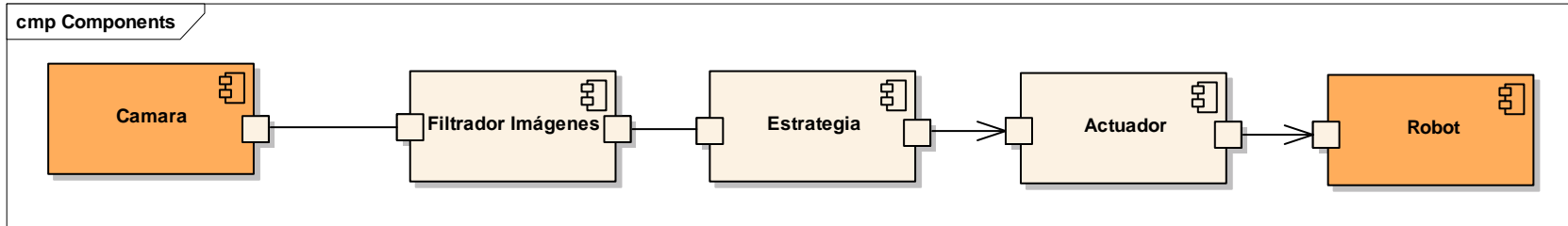
Especificar C&C ...

-Performance
- Primer diseño



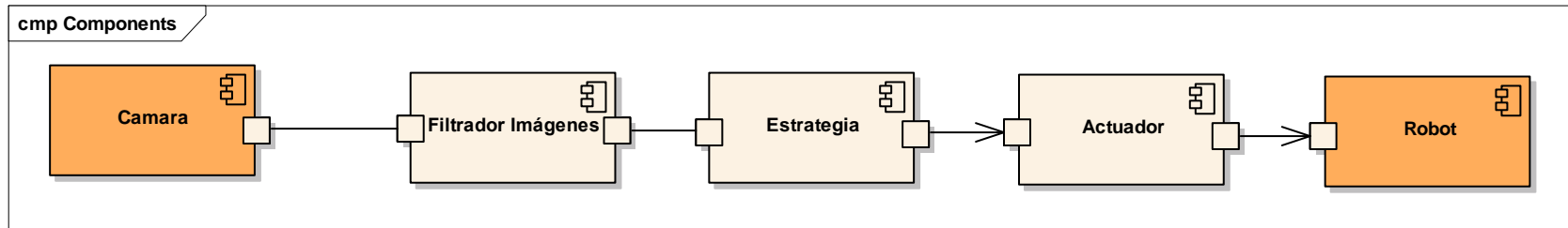
Especificar C&C ...

-Performance **- Primer diseño**



Especificar C&C ...

-Performance **- Primer diseño**



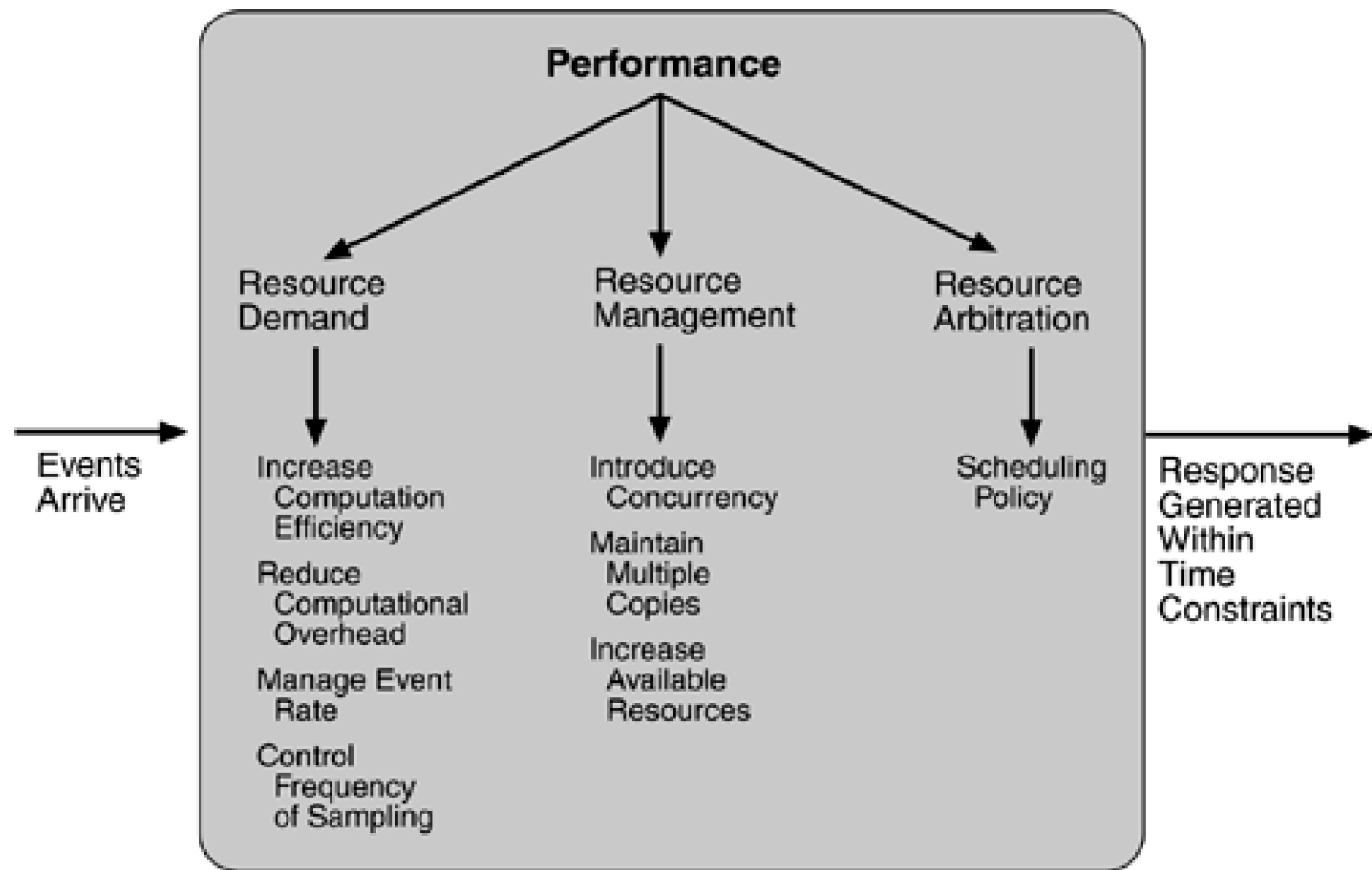
¿Permite cumplir con el escenario?

Especificar C&C ...

-Performance

- Primer diseño

- Tácticas

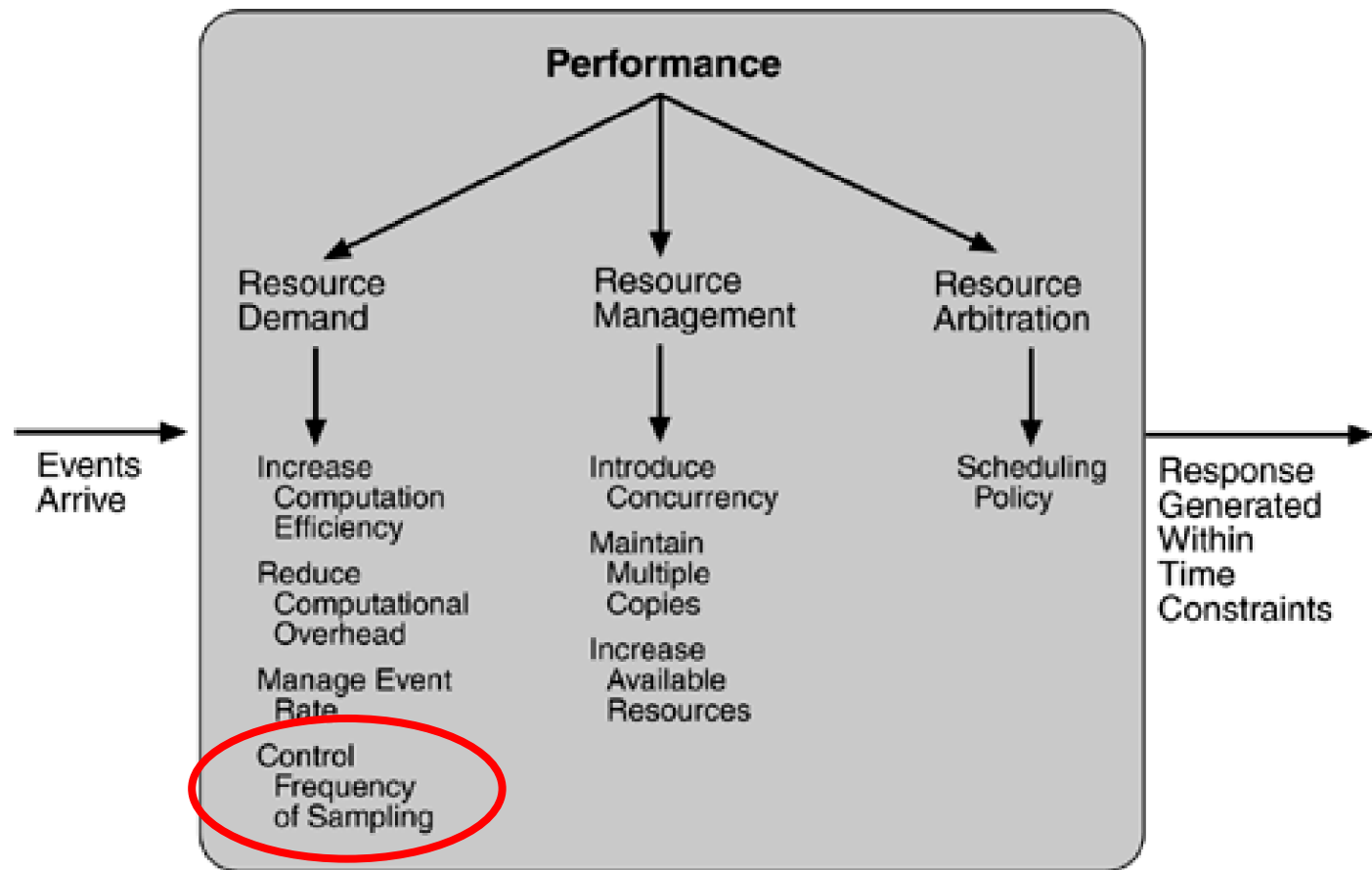


Especificar C&C ...

-Performance

- Primer diseño

- Tácticas

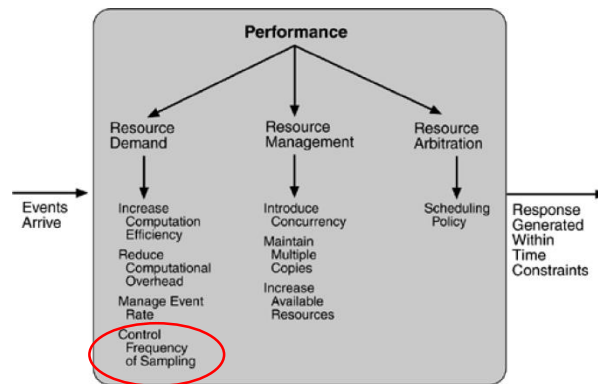


Especificar C&C ...

-Performance

- Primer diseño

-Tácticas



-¿Podemos disminuir la cantidad o descartar imágenes que nos llegan de la cámara?

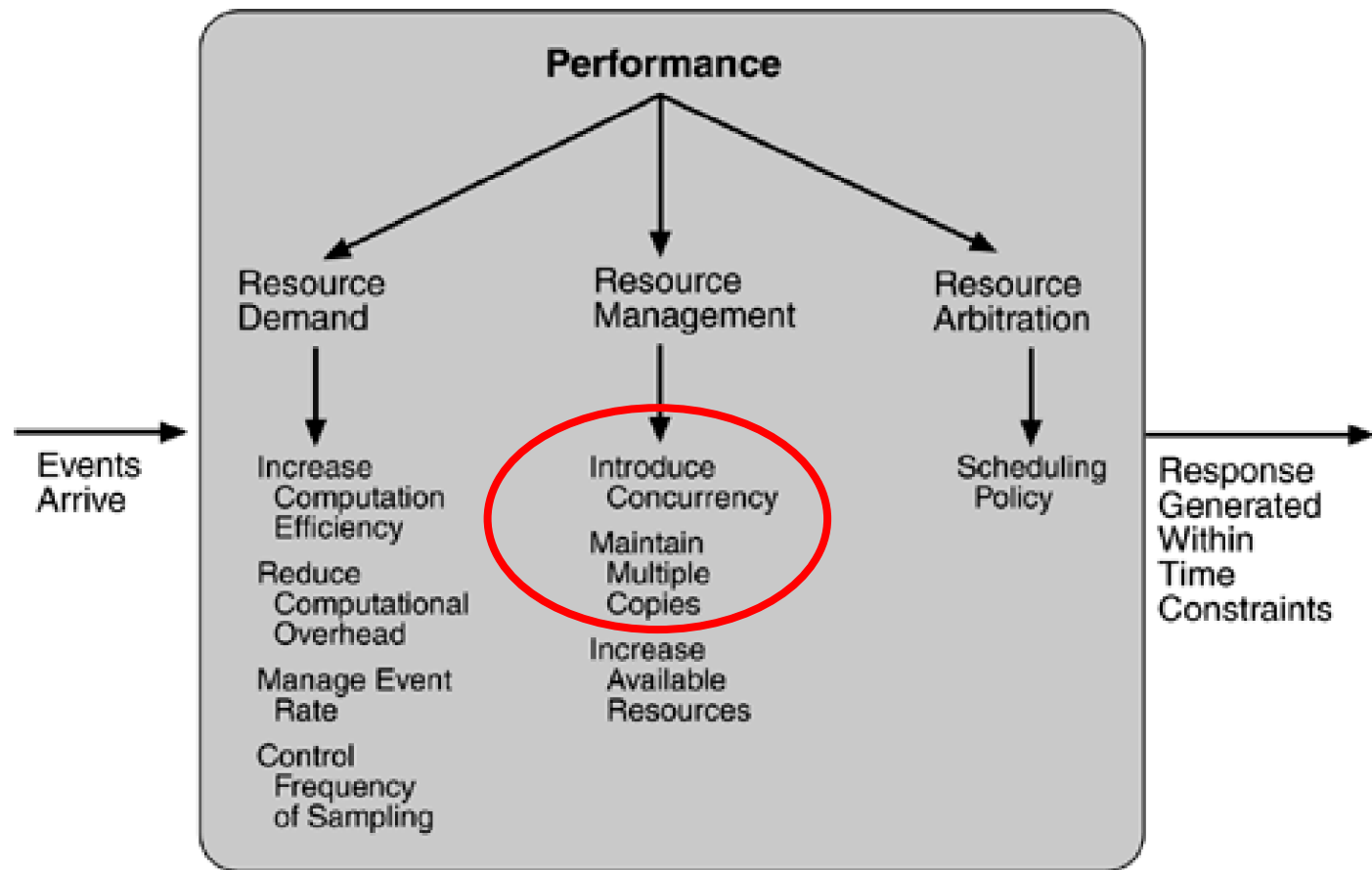
-¿Resuelve nuestro problema?

Especificar C&C ...

-Performance

- *Primer diseño*

- *Tácticas*



Especificar C&C ...

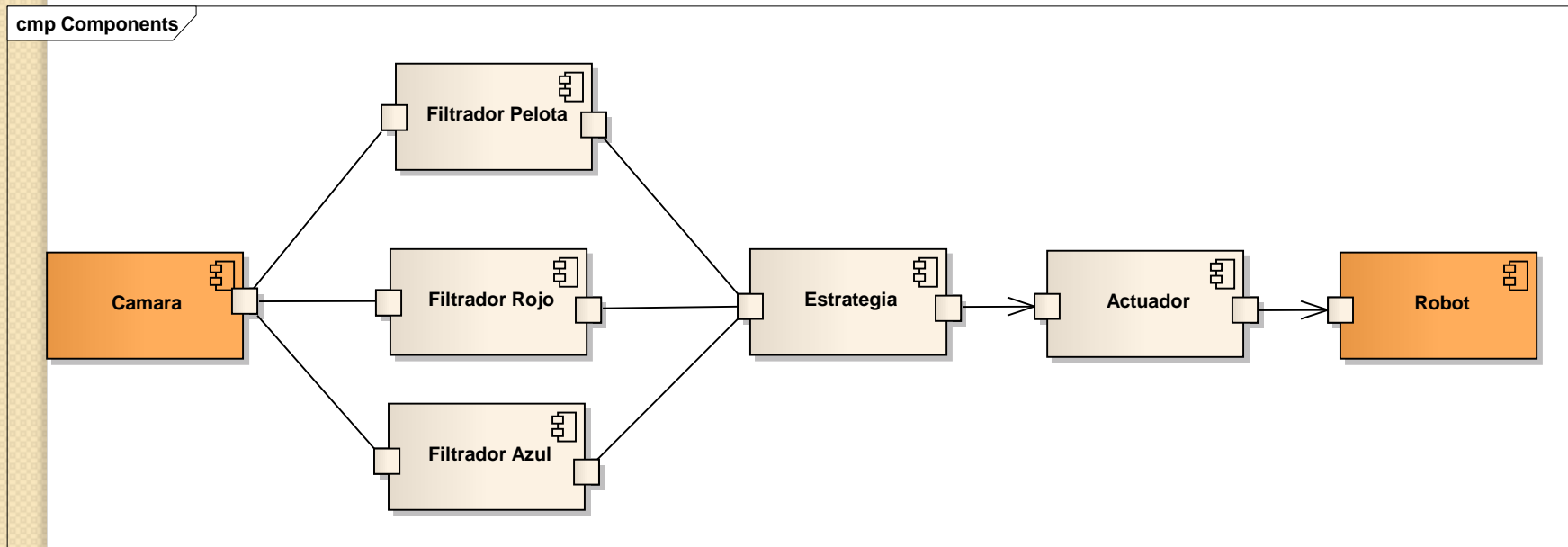
-Performance

- Primer diseño

- Tácticas

-Concurrencia

-Mantener múltiples copias del procesamiento



Y ahora, ¿Permite cumplir con el escenario?

Especificar C&C ...

-Performance

- Primer diseño***

- Tácticas***

 - Concurrencia***

 - Mantener múltiples copias del procesamiento***

Y si no encontramos la pelota con el optimizado...?

Especificar C&C ...

-Performance

- *Primer diseño*

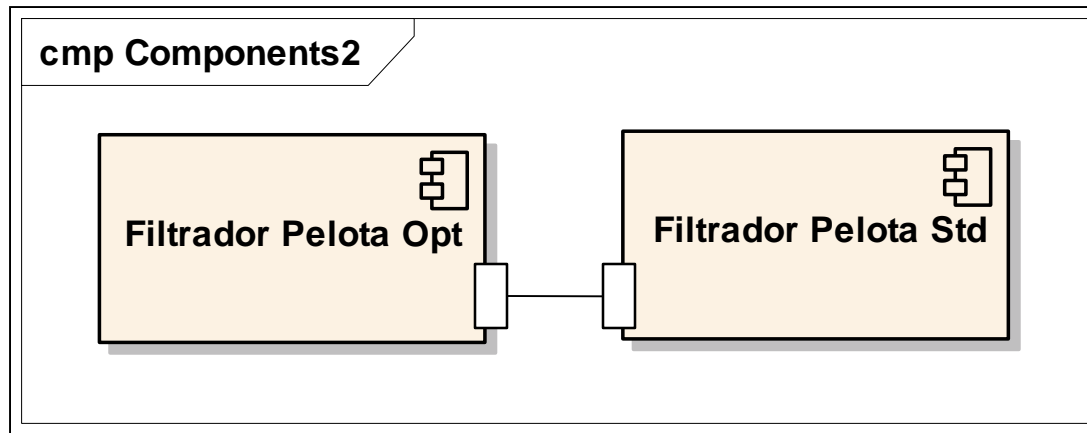
- *Tácticas*

 - *Concurrencia*

 - *Mantener múltiples copias del procesamiento*

Y si no encontramos la pelota con el optimizado...?

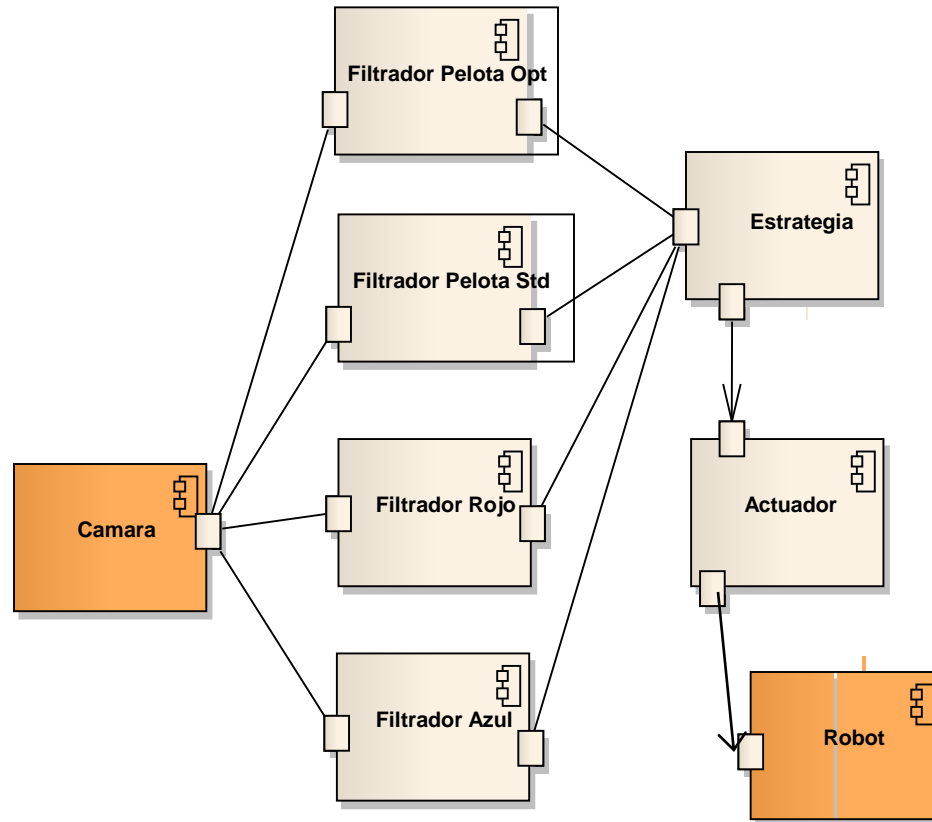
Hagamos zoom en el comp Filtrador Pelota



Puede
demorar
más de 1
segundo

¿Y ahora?

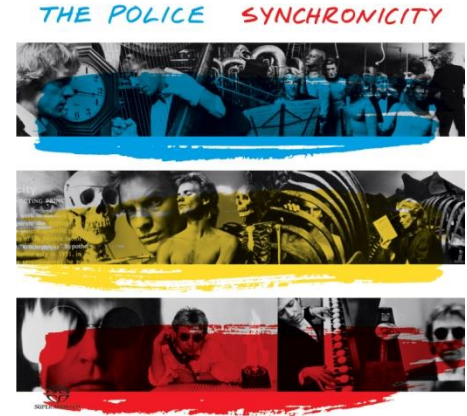
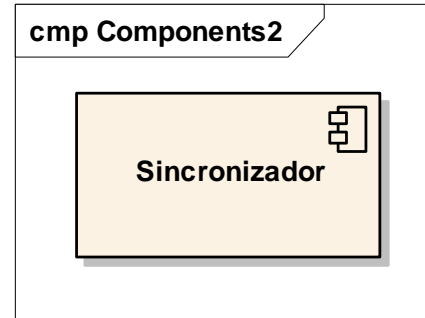
cmp Components



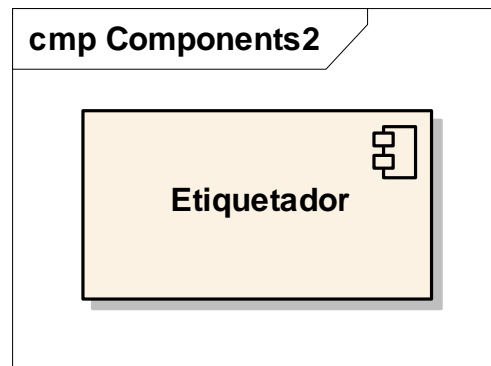
Especificar C&C ...

¿Qué problemas trae la concurrencia?

Sincronización



¿Y en base a qué sincroniza?



Especificar C&C ...

-Performance

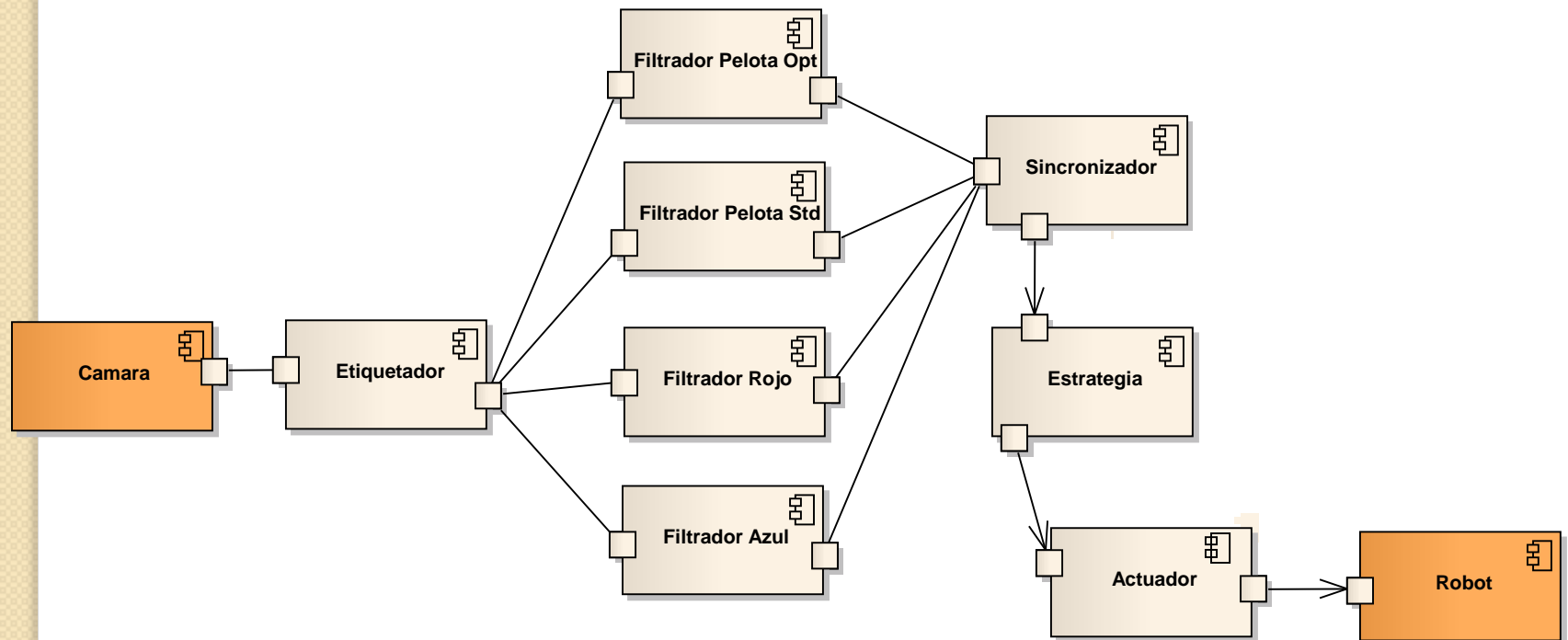
- Primer diseño

- Tácticas

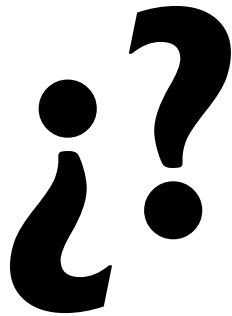
-Concurrencia

-Mantener múltiples copias del procesamiento

cmp Components



Dudas...



Y de yapa, para pensar...

¿Cómo modificarían el C&C para “demostrar” que están cumpliendo el escenario #2 (Detección de Outliers) sin afectar la performance?